

MOTION CONTROLLER



Qseries

SV13/SV22(Motion SFC)

Q173DCPU

Q172DCPU

Programming Manual

● SAFETY PRECAUTIONS ●

(Please read these instructions before using this equipment.)

Before using this product, please read this manual and the relevant manuals introduced in this manual carefully and pay full attention to safety to handle the product correctly.

These precautions apply only to this product. Refer to the Q173DCPU/Q172DCPU Users manual for a description of the Motion controller safety precautions.


In this manual, the safety instructions are ranked as "DANGER" and "CAUTION".



Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.



Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Depending on circumstances, procedures indicated by  CAUTION may also be linked to serious results.

In any case, it is important to follow the directions for usage.

Please save this manual to make it accessible when required and always forward it to the end user.

For Safe Operations

1. Prevention of electric shocks

DANGER

- Never open the front case or terminal covers while the power is ON or the unit is running, as this may lead to electric shocks.
- Never run the unit with the front case or terminal cover removed. The high voltage terminal and charged sections will be exposed and may lead to electric shocks.
- Never open the front case or terminal cover at times other than wiring work or periodic inspections even if the power is OFF. The insides of the Motion controller and servo amplifier are charged and may lead to electric shocks.
- Completely turn off the externally supplied power used in the system before mounting or removing the module, performing wiring work, or inspections. Failing to do so may lead to electric shocks.
- When performing wiring work or inspections, turn the power OFF, wait at least ten minutes, and then check the voltage with a tester, etc.. Failing to do so may lead to electric shocks.
- Be sure to ground the Motion controller, servo amplifier and servomotor. (Ground resistance : 100 Ω or less) Do not ground commonly with other devices.
- The wiring work and inspections must be done by a qualified technician.
- Wire the units after installing the Motion controller, servo amplifier and servomotor. Failing to do so may lead to electric shocks or damage.
- Never operate the switches with wet hands, as this may lead to electric shocks.
- Do not damage, apply excessive stress, place heavy things on or sandwich the cables, as this may lead to electric shocks.
- Do not touch the Motion controller, servo amplifier or servomotor terminal blocks while the power is ON, as this may lead to electric shocks.
- Do not touch the built-in power supply, built-in grounding or signal wires of the Motion controller and servo amplifier, as this may lead to electric shocks.

2. For fire prevention

CAUTION

- Install the Motion controller, servo amplifier, servomotor and regenerative resistor on incombustible. Installing them directly or close to combustibles will lead to fire.
- If a fault occurs in the Motion controller or servo amplifier, shut the power OFF at the servo amplifier's power source. If a large current continues to flow, fire may occur.
- When using a regenerative resistor, shut the power OFF with an error signal. The regenerative resistor may abnormally overheat due to a fault in the regenerative transistor, etc., and may lead to fire.
- Always take heat measures such as flame proofing for the inside of the control panel where the servo amplifier or regenerative resistor is installed and for the wires used. Failing to do so may lead to fire.
- Do not damage, apply excessive stress, place heavy things on or sandwich the cables, as this may lead to fire.

3. For injury prevention

CAUTION

- Do not apply a voltage other than that specified in the instruction manual on any terminal. Doing so may lead to destruction or damage.
- Do not mistake the terminal connections, as this may lead to destruction or damage.
- Do not mistake the polarity (+ / -), as this may lead to destruction or damage.
- Do not touch the heat radiating fins of controller or servo amplifier, regenerative resistor and servomotor, etc., while the power is ON and for a short time after the power is turned OFF. In this timing, these parts become very hot and may lead to burns.
- Always turn the power OFF before touching the servomotor shaft or coupled machines, as these parts may lead to injuries.
- Do not go near the machine during test operations or during operations such as teaching. Doing so may lead to injuries.

4. Various precautions

Strictly observe the following precautions.

Mistaken handling of the unit may lead to faults, injuries or electric shocks.

(1) System structure

CAUTION

- Always install a leakage breaker on the Motion controller and servo amplifier power source.
- If installation of an electromagnetic contactor for power shut off during an error, etc., is specified in the instruction manual for the servo amplifier, etc., always install the electromagnetic contactor.
- Install the emergency stop circuit externally so that the operation can be stopped immediately and the power shut off.
- Use the Motion controller, servo amplifier, servomotor and regenerative resistor with the correct combinations listed in the instruction manual. Other combinations may lead to fire or faults.
- Use the CPU module, base unit and motion module with the correct combinations listed in the instruction manual. Other combinations may lead to faults.
- If safety standards (ex., robot safety rules, etc.,) apply to the system using the Motion controller, servo amplifier and servomotor, make sure that the safety standards are satisfied.
- Construct a safety circuit externally of the Motion controller or servo amplifier if the abnormal operation of the Motion controller or servo amplifier differ from the safety directive operation in the system.
- In systems where coasting of the servomotor will be a problem during the forced stop, emergency stop, servo OFF or power supply OFF, use dynamic brakes.
- Make sure that the system considers the coasting amount even when using dynamic brakes.
- In systems where perpendicular shaft dropping may be a problem during the forced stop, emergency stop, servo OFF or power supply OFF, use both dynamic brakes and electromagnetic brakes.

CAUTION

- The dynamic brakes must be used only on errors that cause the forced stop, emergency stop, or servo OFF. These brakes must not be used for normal braking.
- The brakes (electromagnetic brakes) assembled into the servomotor are for holding applications, and must not be used for normal braking.
- The system must have a mechanical allowance so that the machine itself can stop even if the stroke limits switch is passed through at the max. speed.
- Use wires and cables that have a wire diameter, heat resistance and bending resistance compatible with the system.
- Use wires and cables within the length of the range described in the instruction manual.
- The ratings and characteristics of the parts (other than Motion controller, servo amplifier and servomotor) used in a system must be compatible with the Motion controller, servo amplifier and servomotor.
- Install a cover on the shaft so that the rotary parts of the servomotor are not touched during operation.
- There may be some cases where holding by the electromagnetic brakes is not possible due to the life or mechanical structure (when the ball screw and servomotor are connected with a timing belt, etc.). Install a stopping device to ensure safety on the machine side.

(2) Parameter settings and programming

CAUTION

- Set the parameter values to those that are compatible with the Motion controller, servo amplifier, servomotor and regenerative resistor model and the system application. The protective functions may not function if the settings are incorrect.
- The regenerative resistor model and capacity parameters must be set to values that conform to the operation mode, servo amplifier and servo power supply module. The protective functions may not function if the settings are incorrect.
- Set the mechanical brake output and dynamic brake output validity parameters to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.
- Set the stroke limit input validity parameter to a value that is compatible with the system application. The protective functions may not function if the setting is incorrect.
- Set the servomotor encoder type (increment, absolute position type, etc.) parameter to a value that is compatible with the system application. The protective functions may not function if the setting is incorrect.
- Set the servomotor capacity and type (standard, low-inertia, flat, etc.) parameter to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.
- Set the servo amplifier capacity and type parameters to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.

CAUTION

- Use the program commands for the program with the conditions specified in the instruction manual.
- Set the sequence function program capacity setting, device capacity, latch validity range, I/O assignment setting, and validity of continuous operation during error detection to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.
- Some devices used in the program have fixed applications, so use these with the conditions specified in the instruction manual.
- The input devices and data registers assigned to the link will hold the data previous to when communication is terminated by an error, etc. Thus, an error correspondence interlock program specified in the instruction manual must be used.
- Use the interlock program specified in the intelligent function module's instruction manual for the program corresponding to the intelligent function module.

(3) Transportation and installation

CAUTION

- Transport the product with the correct method according to the mass.
- Use the servomotor suspension bolts only for the transportation of the servomotor. Do not transport the servomotor with machine installed on it.
- Do not stack products past the limit.
- When transporting the Motion controller or servo amplifier, never hold the connected wires or cables.
- When transporting the servomotor, never hold the cables, shaft or detector.
- When transporting the Motion controller or servo amplifier, never hold the front case as it may fall off.
- When transporting, installing or removing the Motion controller or servo amplifier, never hold the edges.
- Install the unit according to the instruction manual in a place where the mass can be withstood.
- Do not get on or place heavy objects on the product.
- Always observe the installation direction.
- Keep the designated clearance between the Motion controller or servo amplifier and control panel inner surface or the Motion controller and servo amplifier, Motion controller or servo amplifier and other devices.
- Do not install or operate Motion controller, servo amplifiers or servomotors that are damaged or that have missing parts.
- Do not block the intake/outtake ports of the Motion controller, servo amplifier and servomotor with cooling fan.
- Do not allow conductive matter such as screw or cutting chips or combustible matter such as oil enter the Motion controller, servo amplifier or servomotor.

⚠ CAUTION

- The Motion controller, servo amplifier and servomotor are precision machines, so do not drop or apply strong impacts on them.
- Securely fix the Motion controller, servo amplifier and servomotor to the machine according to the instruction manual. If the fixing is insufficient, these may come off during operation.
- Always install the servomotor with reduction gears in the designated direction. Failing to do so may lead to oil leaks.
- Store and use the unit in the following environmental conditions.

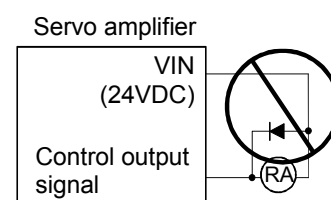
Environment	Conditions	
	Motion controller/Servo amplifier	Servomotor
Ambient temperature	According to each instruction manual.	0°C to +40°C (With no freezing) (32°F to +104°F)
Ambient humidity	According to each instruction manual.	80% RH or less (With no dew condensation)
Storage temperature	According to each instruction manual.	-20°C to +65°C (-4°F to +149°F)
Atmosphere	Indoors (where not subject to direct sunlight). No corrosive gases, flammable gases, oil mist or dust must exist	
Altitude	1000m (3280.84ft.) or less above sea level	
Vibration	According to each instruction manual	

- When coupling with the synchronous encoder or servomotor shaft end, do not apply impact such as by hitting with a hammer. Doing so may lead to detector damage.
- Do not apply a load larger than the tolerable load onto the synchronous encoder and servomotor shaft. Doing so may lead to shaft breakage.
- When not using the module for a long time, disconnect the power line from the Motion controller or servo amplifier.
- Place the Motion controller and servo amplifier in static electricity preventing vinyl bags and store.
- When storing for a long time, please contact with our sales representative.
Also, execute a trial operation.

(4) Wiring

⚠ CAUTION

- Correctly and securely wire the wires. Reconfirm the connections for mistakes and the terminal screws for tightness after wiring. Failing to do so may lead to run away of the servomotor.
- After wiring, install the protective covers such as the terminal covers to the original positions.
- Do not install a phase advancing capacitor, surge absorber or radio noise filter (option FR-BIF) on the output side of the servo amplifier.
- Correctly connect the output side (terminal U, V, W). Incorrect connections will lead the servomotor to operate abnormally.
- Do not connect a commercial power supply to the servomotor, as this may lead to trouble.
- Do not mistake the direction of the surge absorbing diode installed on the DC relay for the control signal output of brake signals, etc. Incorrect installation may lead to signals not being output when trouble occurs or the protective functions not functioning.
- Do not connect or disconnect the connection cables between each unit, the encoder cable or PLC expansion cable while the power is ON.
- Securely tighten the cable connector fixing screws and fixing mechanisms. Insufficient fixing may lead to the cables coming off during operation.
- Do not bundle the power line or cables.



(5) Trial operation and adjustment

⚠ CAUTION

- Confirm and adjust the program and each parameter before operation. Unpredictable movements may occur depending on the machine.
- Extreme adjustments and changes may lead to unstable operation, so never make them.
- When using the absolute position system function, on starting up, and when the Motion controller or absolute value motor has been replaced, always perform a home position return.

(6) Usage methods

CAUTION

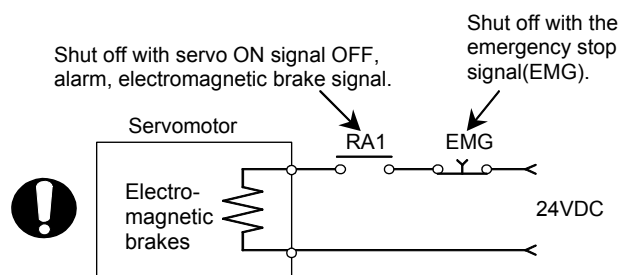
- Immediately turn OFF the power if smoke, abnormal sounds or odors are emitted from the Motion controller, servo amplifier or servomotor.
- Always execute a test operation before starting actual operations after the program or parameters have been changed or after maintenance and inspection.
- Do not attempt to disassemble and repair the units excluding a qualified technician whom our company recognized.
- Do not make any modifications to the unit.
- Keep the effect or electromagnetic obstacles to a minimum by installing a noise filter or by using wire shields, etc. Electromagnetic obstacles may affect the electronic devices used near the Motion controller or servo amplifier.
- When using the CE Mark-compliant equipment, refer to the "EMC Installation Guidelines" (data number IB(NA)-67339) for the Motion controllers and refer to the corresponding EMC guideline information for the servo amplifiers, inverters and other equipment.
- Use the units with the following conditions.

Item	Conditions					
	Q61P-A1	Q61P-A2	Q61P	Q62P	Q63P	Q64P
Input power	100 to 120VAC ^{+10%} _{-15%} (85 to 132VAC)	200 to 240VAC ^{+10%} _{-15%} (170 to 264VAC)	100 to 240VAC ^{+10%} _{-15%} (85 to 264VAC)		24VDC ^{+30%} _{-35%} (15.6 to 31.2VDC)	100 to 120VAC ^{+10%} _{-15%} / 200 to 240VAC ^{+10%} _{-15%} (85 to 132VAC/ 170 to 264VAC)
Input frequency	50/60Hz \pm 5%					
Tolerable momentary power failure	20ms or less					

(7) Corrective actions for errors

⚠ CAUTION

- If an error occurs in the self diagnosis of the Motion controller or servo amplifier, confirm the check details according to the instruction manual, and restore the operation.
- If a dangerous state is predicted in case of a power failure or product failure, use a servomotor with electromagnetic brakes or install a brake mechanism externally.
- Use a double circuit construction so that the electromagnetic brake operation circuit can be operated by emergency stop signals set externally.



- If an error occurs, remove the cause, secure the safety and then resume operation after alarm release.
- The unit may suddenly resume operation after a power failure is restored, so do not go near the machine. (Design the machine so that personal safety can be ensured even if the machine restarts suddenly.)

(8) Maintenance, inspection and part replacement

⚠ CAUTION

- Perform the daily and periodic inspections according to the instruction manual.
- Perform maintenance and inspection after backing up the program and parameters for the Motion controller and servo amplifier.
- Do not place fingers or hands in the clearance when opening or closing any opening.
- Periodically replace consumable parts such as batteries according to the instruction manual.
- Do not touch the lead sections such as ICs or the connector contacts.
- Before touching the module, always touch grounded metal, etc. to discharge static electricity from human body. Failure to do so may cause the module to fail or malfunction.
- Do not directly touch the module's conductive parts and electronic components. Touching them could cause an operation failure or give damage to the module.
- Do not place the Motion controller or servo amplifier on metal that may cause a power leakage or wood, plastic or vinyl that may cause static electricity buildup.
- Do not perform a megger test (insulation resistance measurement) during inspection.

CAUTION

- When replacing the Motion controller or servo amplifier, always set the new module settings correctly.
- When the Motion controller or absolute value motor has been replaced, carry out a home position return operation using one of the following methods, otherwise position displacement could occur.
 - 1) After writing the servo data to the Motion controller using programming software, switch on the power again, then perform a home position return operation.
 - 2) Using the backup function of the programming software, load the data backed up before replacement.
- After maintenance and inspections are completed, confirm that the position detection of the absolute position detector function is correct.
- Do not drop or impact the battery installed to the module.

Doing so may damage the battery, causing battery liquid to leak in the battery. Do not use the dropped or impacted battery, but dispose of it.
- Do not short circuit, charge, overheat, incinerate or disassemble the batteries.
- The electrolytic capacitor will generate gas during a fault, so do not place your face near the Motion controller or servo amplifier.
- The electrolytic capacitor and fan will deteriorate. Periodically replace these to prevent secondary damage from faults. Replacements can be made by our sales representative.

(9) About processing of waste

When you discard Motion controller, servo amplifier, a battery (primary battery) and other option articles, please follow the law of each country (area).

CAUTION

- This product is not designed or manufactured to be used in equipment or systems in situations that can affect or endanger human life.
- When considering this product for operation in special applications such as machinery or systems used in passenger transportation, medical, aerospace, atomic power, electric power, or submarine repeating applications, please contact your nearest Mitsubishi sales representative.
- Although this product was manufactured under conditions of strict quality control, you are strongly advised to install safety devices to forestall serious accidents when it is used in facilities where a breakdown in the product is likely to cause a serious accident.

(10) General cautions

CAUTION

- All drawings provided in the instruction manual show the state with the covers and safety partitions removed to explain detailed sections. When operating the product, always return the covers and partitions to the designated positions, and operate according to the instruction manual.

REVISIONS

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	Revision
Jan., 2008	IB(NA)-0300135-A	First edition

Japanese Manual Number IB(NA)-0300127

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

INTRODUCTION

Thank you for choosing the Mitsubishi Motion controller Q173DCPU/Q172DCPU.
Before using the equipment, please read this manual carefully to develop full familiarity with the functions and performance of the Motion controller you have purchased, so as to ensure correct use.

CONTENTS

Safety Precautions	A- 1
Revisions	A-11
Contents	A-12
About Manuals	A-17

1. OVERVIEW	1- 1 to 1-36
-------------	--------------

1.1 Overview	1- 1
1.2 Features	1- 3
1.2.1 Features of Motion SFC programs	1- 3
1.2.2 Performance specifications	1- 4
1.2.3 Operation control/transition control specifications	1- 6
1.2.4 Positioning dedicated devices	1-10

2. STRUCTURE OF THE MOTION CPU PROGRAM	2- 1 to 2- 4
--	--------------

2.1 Motion Control in SV13/SV22 Real Mode	2- 2
2.2 Motion Control in SV22 Virtual Mode	2- 3

3. MOTION DEDICATED PLC INSTRUCTION	3- 1 to 3-62
-------------------------------------	--------------

3.1 Outline of Motion Dedicated PLC Instruction	3- 1
3.2 Motion Dedicated PLC Instruction	3- 1
3.2.1 Motion SFC start request from the PLC CPU to the Motion CPU: D(P).SFCS (PLC instruction: D(P).SFCS)	3- 2
3.2.2 Servo program start request from the PLC CPU to the Motion CPU: D(P).SVST (PLC instruction: D(P).SVST)	3- 6
3.2.3 Current value change instruction from the PLC CPU to the Motion CPU: D(P).CHGA (PLC instruction: D(P).CHGA)	3-12
3.2.4 Speed change instruction from the PLC CPU to the Motion CPU: D(P).CHGV (PLC instruction: D(P).CHGV)	3-26
3.2.5 Torque limit value change request instruction from the PLC CPU to the Motion CPU: D(P).CHGT (PLC instruction: D(P).CHGT)	3-32
3.2.6 Write device data of the self CPU to the device of other CPU: D(P).DDWR (PLC instruction: D(P).DDWR)	3-37
3.2.7 Read device data of other CPU to the device of self CPU D(P).DDRD (PLC instruction: D(P).DDRD)	3-42
3.2.8 Interrupt instruction to the other CPU: D(P).GINT (PLC instruction: D(P).GINT)	3-47
3.3 Precautions	3-51

4. MOTION SFC PROGRAMS

4- 1 to 4-28

4.1 Motion SFC Program Configuration	4- 1
4.2 Motion SFC Chart Symbol List	4- 2
4.3 Branch and Coupling Chart List.....	4- 5
4.4 Motion SFC Program Name	4- 9
4.5 Steps.....	4-10
4.5.1 Motion control step	4-10
4.5.2 Operation control step.....	4-11
4.5.3 Subroutine call/start step.....	4-12
4.5.4 Clear step	4-14
4.6 Transitions	4-15
4.7 Jump, Pointer	4-18
4.8 END	4-18
4.9 Branches, Couplings	4-19
4.9.1 Series transition.....	4-19
4.9.2 Selective branch, selective coupling.....	4-20
4.9.3 Parallel branch, parallel coupling	4-21
4.10 Y/N Transitions.....	4-23
4.11 Motion SFC Comments	4-27

5. OPERATION CONTROL PROGRAMS

5- 1 to 5-96

5.1 Operation Control Programs.....	5- 1
5.2 Device Descriptions	5- 7
5.3 Constant Descriptions	5-10
5.4 Binary Operations	5-11
5.4.1 Substitution : =.....	5-11
5.4.2 Addition : +.....	5-13
5.4.3 Subtraction : -	5-14
5.4.4 Multiplication : *	5-15
5.4.5 Division : /	5-16
5.4.6 Remainder : %.....	5-17
5.5 Bit Operations.....	5-18
5.5.1 Bit inversion(Complement) : ~	5-18
5.5.2 Bit logical AND : &	5-19
5.5.3 Bit logical OR : 	5-20
5.5.4 Bit exclusive logical OR : ^.....	5-21
5.5.5 Bit right shift : >>	5-22
5.5.6 Bit left shift : <<	5-23
5.5.7 Sign inversion(Complement of 2) : -	5-24
5.6 Standard Functions	5-25
5.6.1 Sine : SIN.....	5-25
5.6.2 Cosine : COS.....	5-26
5.6.3 Tangent : TAN	5-27
5.6.4 Arcsine : ASIN	5-28
5.6.5 Arccosine : ACOS	5-29
5.6.6 Arctangent : ATAN	5-30
5.6.7 Square root : SQRT	5-31

5.6.8 Natural logarithm : LN	5-32
5.6.9 Exponential operation : EXP	5-33
5.6.10 Absolute value : ABS	5-34
5.6.11 Round-off : RND	5-35
5.6.12 Round-down : FIX	5-36
5.6.13 Round-up : FUP	5-37
5.6.14 BCD → BIN conversion : BIN	5-38
5.6.15 BIN → BCD conversion : BCD	5-39
5.7 Type Conversions	5-40
5.7.1 Signed 16-bit integer value conversion : SHORT	5-40
5.7.2 Unsigned 16-bit integer value conversion : USHORT	5-41
5.7.3 Signed 32-bit integer value conversion : LONG	5-42
5.7.4 Unsigned 32-bit integer value conversion : ULONG	5-43
5.7.5 Signed 64-bit floating-point value conversion : FLOAT	5-44
5.7.6 Unsigned 64-bit floating-point value conversion : UFLOAT	5-45
5.8 Bit Device Statuses	5-46
5.8.1 ON (Normally open contact) : (None)	5-46
5.8.2 OFF (Normally closed contact) : !	5-47
5.9 Bit Device Controls	5-48
5.9.1 Device set : SET	5-48
5.9.2 Device reset : RST	5-50
5.9.3 Device output : DOUT	5-52
5.9.4 Device input : DIN	5-53
5.9.5 Bit device output : OUT	5-54
5.10 Logical Operations	5-55
5.10.1 Logical acknowledgement : (None)	5-55
5.10.2 Logical negation : !	5-56
5.10.3 Logical AND : *	5-57
5.10.4 Logical OR : +	5-58
5.11 Comparison Operations	5-59
5.11.1 Equal to : ==	5-59
5.11.2 Not equal to : !=	5-60
5.11.3 Less than : <	5-61
5.11.4 Less than or equal to : <=	5-62
5.11.5 More than : >	5-63
5.11.6 More than or equal to : >=	5-64
5.12 Motion-Dedicated Functions(CHGV, CHGT)	5-65
5.12.1 Speed change request : CHGV	5-65
5.12.2 Torque limit value change request : CHGT	5-71
5.13 Other Instructions	5-73
5.13.1 Event task enable : EI	5-73
5.13.2 Event task disable : DI	5-74
5.13.3 No operation : NOP	5-75
5.13.4 Block transfer : BMOV	5-76
5.13.5 Same data block transfer : FMOV	5-79
5.13.6 Write device data to CPU shared memory of the self CPU : MULTW	5-81
5.13.7 Read device data from CPU shared memory of the other CPU : MULTR	5-84
5.13.8 Write device data to intelligent function module : TO	5-87
5.13.9 Read device data from intelligent function module : FROM	5-90

5.13.10 Time to wait : TIME	5-93
5.14 Comment Statement : //	5-95

6. TRANSITION PROGRAMS	6- 1 to 6- 2
-------------------------------	---------------------

6.1 Transition Programs.....	6- 1
------------------------------	------

7. MOTION CONTROL PROGRAMS	7- 1 to 7-22
-----------------------------------	---------------------

7.1 Servo Instruction List.....	7- 1
7.2 Servomotor/Virtual Servomotor Shaft Current Value Change.....	7-14
7.3 Synchronous Encoder Shaft Current Value Change Control (SV22 only).....	7-17
7.4 Cam Shaft Within-One-Revolution Current Value Change Control (SV22 only).....	7-20
7.5 Programming Instructions.....	7-22
7.5.1 Cancel • start	7-22
7.5.2 Indirect designation using motion devices.....	7-22

8. MOTION DEVICES	8- 1 to 8- 4
--------------------------	---------------------

8.1 Motion Registers (#0 to #8735)	8- 1
8.2 Coasting Timer (FT).....	8- 4

9. OPERATION FOR MOTION SFC AND PARAMETER	9- 1 to 9-18
--	---------------------

9.1 Task Definitions.....	9- 1
9.2 Number of Consecutive Transitions and Task Operation	9- 1
9.2.1 Number of consecutive transitions	9- 1
9.2.2 Task operation.....	9- 2
9.3 Execution Status of The Multiple Task.....	9- 6
9.4 How to Start The Motion SFC Program	9- 7
9.4.1 Automatic start.....	9- 7
9.4.2 Start from the Motion SFC program	9- 7
9.4.3 Start from PLC (PLC instruction D(P).SFCS)	9- 7
9.5 How to End The Motion SFC Program.....	9- 7
9.6 How to Change from One Motion SFC Program to Another	9- 8
9.7 Operation Performed at Multiple CPU System Power-Off or Reset	9- 8
9.8 Operation Performed when CPU is Switched from RUN/STOP	9- 8
9.9 Operation Performed when PLC Ready Flag (M2000) Turns OFF/ON.....	9- 9
9.10 Operation at The Error Occurrence.....	9- 9
9.11 Task Parameters.....	9-10
9.12 Program Parameters.....	9-12

10.ONLINE CHANGE IN THE MOTION SFC PROGRM	10- 1 to 10- 8
--	-----------------------

10.1 Online Change in The Motion SFC Program.....	10- 1
10.1.1 Operating method for the online change.....	10- 3
10.1.2 Transfer of program	10- 6

About Manuals

The following manuals are also related to this product.

In necessary, order them by quoting the details in the tables below.

Related Manuals

(1) Motion controller

Manual Name	Manual Number (Model Code)
<p>Q173DCPU/Q172DCPU Motion controller User's Manual</p> <p>This manual explains specifications of the Motion CPU modules, Q172DLX Servo external signal interface module, Q172DEX Synchronous encoder interface module, Q173DPX Manual pulse generator interface module, Power supply modules, Servo amplifiers, SSCNETⅢ cables, Synchronous encoder cables and others.</p> <p>(Optional)</p>	<p>IB-0300133 (1XB927)</p>
<p>Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)</p> <p>This manual explains the Multiple CPU system configuration, performance specifications, common parameters, auxiliary/applied functions, error lists and others.</p> <p>(Optional)</p>	<p>IB-0300134 (1XB928)</p>
<p>Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)</p> <p>This manual explains the servo parameters, positioning instructions, device lists, error lists and others.</p> <p>(Optional)</p>	<p>IB-0300136 (1XB930)</p>
<p>Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)</p> <p>This manual explains the dedicated instructions to use the synchronous control by virtual main shaft, mechanical system program create mechanical module, servo parameters, positioning instructions, device lists, error lists and others.</p> <p>(Optional)</p>	<p>IB-0300137 (1XB931)</p>

(2) PLC

Manual Name	Manual Number (Model Code)
QCPU User's Manual (Hardware Design, Maintenance and Inspection) This manual explains the specifications of the QCPU modules, power supply modules, base modules, extension cables, memory card battery and others. (Optional)	SH-080483ENG (13JR73)
QCPU User's Manual (Function Explanation, Program Fundamentals) This manual explains the functions, programming methods and devices and others to create programs with the QCPU. (Optional)	SH-080484ENG (13JR74)
QCPU User's Manual (Multiple CPU System) This manual explains the functions, programming methods and cautions and others to construct the Multiple CPU system with the QCPU. (Optional)	SH-080485ENG (13JR75)
QCPU (Q Mode)/QnACPU Programming Manual (Common Instructions) This manual explains how to use the sequence instructions, basic instructions, application instructions and micro computer program. (Optional)	SH-080039 (13JF58)
QCPU (Q Mode)/QnACPU Programming Manual (PID Control Instructions) This manual explains the dedicated instructions used to exercise PID control. (Optional)	SH-080040 (13JF59)
QCPU (Q Mode)/QnACPU Programming Manual (SFC) This manual explains the system configuration, performance specifications, functions, programming, debugging, error codes and others of MELSAP3. (Optional)	SH-080041 (13JF60)
I/O Module Type Building Block User's Manual This manual explains the specifications of the I/O modules, connector, connector/terminal block conversion modules and others. (Optional)	SH-080042 (13JL99)

(3) Servo amplifier

Manual Name	Manual Number (Model Code)
MR-J3-□B Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for MR-J3-□B Servo amplifier. (Optional)	SH-030051 (1CW202)
Fully Closed Loop Control MR-J3-□B-RJ006 Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for Fully Closed Loop Control MR-J3-□B-RJ006 Servo amplifier. (Optional)	SH-030056 (1CW304)

1. OVERVIEW

1.1 Overview

This programming manual describes the Motion SFC program of the operating system software "SW8DNC-SV13Q□", "SW8DNC-SV22Q□" for Motion CPU module (Q173DCPU/Q172DCPU).

In this manual, the following abbreviations are used.

Generic term/Abbreviation	Description
Q173DCPU/Q172DCPU or Motion CPU (module)	Q173DCPU/Q172DCPU Motion CPU module
Q172DLX/Q172DEX/Q173DPX or Motion module	Q172DLX Servo external signals interface module/ Q172DEX Serial Synchronous encoder interface module ^(Note-1) / Q173DPX Manual pulse generator interface module
MR-J3-□B	Servo amplifier model MR-J3-□B
AMP or Servo amplifier	General name for "Servo amplifier model MR-J3-□B"
QCPU, PLC CPU or PLC CPU module	QnUD(H)CPU
Multiple CPU system or Motion system	Abbreviation for "Multiple PLC system of the Q series"
CPU _n	Abbreviation for "CPU No.n (n= 1 to 4) of the CPU module for the Multiple CPU system"
Self CPU	Motion CPU being programmed by the currently open MT Developer project
Programming software package	General name for MT Developer/GX Developer/MR Configurator
Operating system software	General name for "SW8DNC-SV□Q□"
SV13	Operating system software for conveyor assembly use (Motion SFC) : SW8DNC -SV13Q□
SV22	Operating system software for automatic machinery use (Motion SFC) : SW8DNC -SV22Q□
MT Developer	Abbreviation for "Motion controller programming software MT Developer2 (Version 1.00A or later)"
GX Developer	Abbreviation for "MELSEC PLC programming software package GX Developer (Version 8.48A or later)"
MR Configurator	Abbreviation for "Servo setup software package MR Configurator (Version C0 or later)"
Manual pulse generator or MR-HDP01	Abbreviation for "Manual pulse generator (MR-HDP01)"
Serial absolute synchronous encoder or Q170ENC	Abbreviation for "Serial absolute synchronous encoder (Q170ENC)"
SSCNET ^{III} ^(Note-2)	High speed synchronous network between Motion controller and servo amplifier
Absolute position system	General name for "system using the servomotor and servo amplifier for absolute position"
Battery holder unit	Battery holder unit (Q170DBATC)
External battery	General name for "Q170DBATC" and "Q6BAT"
Intelligent function module	Abbreviation for "MELSECNET/H module/Ethernet module/CC-Link module/ Serial communication module"

(Note-1) : Q172DEX can be used in SV22.

(Note-2) : SSCNET: Servo System Controller NETwork

1 OVERVIEW

REMARK

For information about the each module, design method for program and parameter, refer to the following manuals relevant to each module.

Item		Reference Manual
Motion CPU module/Motion unit		Q173DCPU/Q172DCPU User's Manual
PLC CPU, peripheral devices for PLC program design, I/O modules and intelligent function module		Manual relevant to each module
Operation method for MT Developer		Help of each software
SV13/SV22/	<ul style="list-style-type: none">• Multiple CPU system configuration• Performance specification• Design method for common parameter• Auxiliary and applied functions (common)	Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)
	<ul style="list-style-type: none">• Design method for positioning control program in the real mode• Design method for positioning control parameter	Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)
SV22 (Virtual mode)	<ul style="list-style-type: none">• Design method for mechanical system program	Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)

1.2 Features

The Motion CPU and Motion SFC program have the following features.

1.2.1 Features of Motion SFC programs

- (1) Since a program intelligible for anyone can be created in flow chart form by making a sequence of machine operation correspond to each operation step, maintenance nature improves.
- (2) Since transition conditions are judged with Motion CPU side and positioning starts, there is not dispersion in the response time influenced by PLC scan time.
- (3) High speed and high response processing is realizable with the step processing method (only active steps) of Motion SFC.
- (4) Not only positioning control but also numerical operations, device SET/RST, etc. can be processed with Motion CPU side, making via PLC CPU is unnecessary and a tact time can be shortened.
- (5) By transition condition description peculiar to Motion SFC, the instructions to servo amplifier is possible at completion of starting condition.
- (6) By transition condition description peculiar to Motion SFC, after starting, transition to next step is possible without waiting for positioning completion.
- (7) Motion SFC program that responds and executes it at high speed for interrupt input from external source can be set.
- (8) Motion SFC program executed in the fixed cycle (0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms) by synchronizing to the Motion operation cycle can be set.

1 OVERVIEW

1.2.2 Performance specifications

(1) Basic specifications of Q173DCPU/Q172DCPU (a) Motion control specifications

Item		Q173DCPU	Q172DCPU
Number of control axes		Up to 32 axes	Up to 8 axes
Operation cycle (default)	SV13	0.44ms/ 1 to 6 axes 0.88ms/ 7 to 18 axes 1.77ms/19 to 32 axes	0.44ms/ 1 to 6 axes 0.88ms/ 7 to 8 axes
	SV22	0.44ms/ 1 to 4 axes 0.88ms/ 5 to 12 axes 1.77ms/13 to 28 axes 3.55ms/29 to 32 axes	0.44ms/ 1 to 4 axes 0.88ms/ 5 to 8 axes
Interpolation functions		Linear interpolation (Up to 4 axes), Circular interpolation (2 axes), Helical interpolation (3 axes)	
Control modes		PTP(Point to Point) control, Speed control, Speed-position control, Fixed-pitch feed, Constant speed control, Position follow-up control, Speed control with fixed position stop, Speed switching control, High-speed oscillation control, Synchronous control (SV22)	
Acceleration/ deceleration control		Automatic trapezoidal acceleration/deceleration, S-curve acceleration/deceleration	
Compensation		Backlash compensation, Electronic gear, Phase compensation (SV22)	
Programming language		Motion SFC, Dedicated instruction, Mechanical support language (SV22)	
Servo program capacity		14k steps	
Number of positioning points		3200 points (Positioning data can be designated indirectly)	
Peripheral I/F		Via PLC CPU (USB/RS-232)	
Home position return function		Proximity dog type (2 types), Count type (3 types), Data set type (2 types), Dog cradle type, Stopper type (2 types), Limit switch combined type (Home position return re-try function provided, home position shift function provided)	
JOG operation function		Provided	
Manual pulse generator operation function		Possible to connect 3 modules	
Synchronous encoder operation function		Possible to connect 12 modules	Possible to connect 8 modules
M-code function		M-code output function provided M-code completion wait function provided	
Limit switch output function		Number of output points 32 points Watch data: Motion control data/Word device	
Absolute position system		Made compatible by setting battery to servo amplifier. (Possible to select the absolute data method or incremental method for each axis)	
Number of SSCNET III systems (Note-1)		2 systems	1 system

1 OVERVIEW

Motion control specifications (continued)

Item	Q173DCPU	Q172DCPU
Motion related interface module	Q172DLX : 4 modules usable Q172DEX : 6 modules usable Q173DPX : 4 modules usable ^(Note-2)	Q172DLX : 1 module usable Q172DEX : 4 modules usable Q173DPX : 3 modules usable ^(Note-2)

(Note-1) : The servo amplifiers for SSCNET cannot be used.

(Note-2) : When using the incremental synchronous encoder (SV22 use), you can use above number of modules.

When connecting the manual pulse generator, you can use only 1 module.

When connecting the manual pulse generator, you can use only 1 module.

(b) Motion SFC Performance Specifications

Item			Q173DCPU/Q172DCPU	
Motion SFC program capacity	Code total (Motion SFC chart + Operation control + Transition)		543k bytes	
	Text total (Operation control + Transition)		484k bytes	
Motion SFC program	Number of Motion SFC programs		256 (No.0 to 255)	
	Motion SFC chart size/program		Up to 64k bytes (Included Motion SFC chart comments)	
	Number of Motion SFC steps/program		Up to 4094 steps	
	Number of selective branches/branch		255	
	Number of parallel branches/branch		255	
	Parallel branch nesting		Up to 4 levels	
Operation control program (F/FS) / Transition program (G)	Number of operation control programs		4096 with F(Once execution type) and FS(Scan execution type) combined. (F/FS0 to F/FS4095)	
	Number of transition programs		4096(G0 to G4095)	
	Code size/program		Up to approx. 64k bytes (32766 steps)	
	Number of blocks(line)/program		Up to 8192 blocks (in the case of 4 steps(min)/blocks)	
	Number of characters/block		Up to 128 (comment included)	
	Number of operand/block		Up to 64 (operand: constants, word device, bit devices)	
	() nesting/block		Up to 32 levels	
	Descriptive expression	Operation control program	Calculation expression/bit conditional expression	
		Transition program	Calculation expression/bit conditional expression/ comparison conditional expression	
Execute specification	Number of multi execute programs		Up to 256	
	Number of multi active steps		Up to 256 steps/all programs	
	Executed task	Normal task		Execute in main cycle of Motion CPU
		Event task (Execution can be masked.)	Fixed cycle	Execute in fixed cycle (0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms)
			External interrupt	Execute when input ON is set among interrupt module QI60 (16 points).
			PLC interrupt	Execute with interrupt instruction (D(P).GINT) from PLC CPU.
		NMI task		Execute when input ON is set among interrupt module QI60 (16 points).

1 OVERVIEW

1.2.3 Operation control/transition control specifications

(1) Table of the operation control/transition control specifications

Item	Specifications							Remark																																																																					
Expression	Calculation expression		Returns a numeric result. Expressions for calculating indirectly specified data using constants and word devices.					D100+1,SIN(D100), etc.																																																																					
	Conditional expression	Bit conditional expression	Returns a true or false result. Expression for judging ON or OFF of bit device.					M0, !M0, M1*M0, (M1+M2)*(!M3+M4), etc.																																																																					
		Comparison conditional expression	Expressions for comparing indirectly specified data and calculation expressions using constants and word devices.					D100==100 D10<D102+D10, etc.																																																																					
Bit devices	<table><thead><tr><th colspan="2" rowspan="2">Device</th><th rowspan="2">Symbol</th><th colspan="2">Accessibility</th><th colspan="3">Usable tasks</th><th rowspan="2">Description example</th></tr><tr><th>Read</th><th>Write</th><th>Normal</th><th>Event</th><th>NMI</th></tr></thead><tbody><tr><td rowspan="2">Input</td><td>Input module non-loaded range</td><td>X</td><td>○</td><td>○</td><td rowspan="10">○</td><td rowspan="10">○</td><td rowspan="10">○</td><td>X100</td></tr><tr><td>Input module loaded range</td><td>PX</td><td>○</td><td>×</td><td>PX180</td></tr><tr><td rowspan="2">Output</td><td>Output module non-loaded range</td><td>Y</td><td>○</td><td>○</td><td>Y100</td></tr><tr><td>Output module loaded range</td><td>PY</td><td>○</td><td>○</td><td>PY1E0</td></tr><tr><td colspan="2">Internal relay</td><td>M</td><td>○</td><td>○</td><td>M20</td></tr><tr><td colspan="2">Multiple CPU area device</td><td>U□\G□.□</td><td>○</td><td>○</td><td>U3E0\G10200.A</td></tr><tr><td colspan="2">Link relay</td><td>B</td><td>○</td><td>○</td><td>B3FF</td></tr><tr><td colspan="2">Annunciator</td><td>F</td><td>○</td><td>○</td><td>F0</td></tr><tr><td colspan="2">Special relay</td><td>SM</td><td>○</td><td>○</td><td>SM0</td></tr></tbody></table> <p>○ : usable × : unusable</p> <div>CAUTION</div> <p><Restrictions on write-enabled bit devices></p> <p>1) Write to device X is allowed only within the input module non-installed range.</p> <p>2) Special relay has predetermined applications in the system. Do not perform write to other than the user setting device.</p>							Device		Symbol	Accessibility		Usable tasks			Description example	Read	Write	Normal	Event	NMI	Input	Input module non-loaded range	X	○	○	○	○	○	X100	Input module loaded range	PX	○	×	PX180	Output	Output module non-loaded range	Y	○	○	Y100	Output module loaded range	PY	○	○	PY1E0	Internal relay		M	○	○	M20	Multiple CPU area device		U□\G□.□	○	○	U3E0\G10200.A	Link relay		B	○	○	B3FF	Annunciator		F	○	○	F0	Special relay		SM	○	○	SM0	The input X/output Y are written with the actual input PX/actual output PY. It does the layout of the I/O numbers of PX, PY by a set up of as system. (In the operation control program/transition program, automatically represented as PX/PY according to the system setting information.)
Device		Symbol	Accessibility		Usable tasks						Description example																																																																		
			Read	Write	Normal	Event	NMI																																																																						
Input	Input module non-loaded range	X	○	○	○	○	○	X100																																																																					
	Input module loaded range	PX	○	×				PX180																																																																					
Output	Output module non-loaded range	Y	○	○				Y100																																																																					
	Output module loaded range	PY	○	○				PY1E0																																																																					
Internal relay		M	○	○				M20																																																																					
Multiple CPU area device		U□\G□.□	○	○				U3E0\G10200.A																																																																					
Link relay		B	○	○				B3FF																																																																					
Annunciator		F	○	○				F0																																																																					
Special relay		SM	○	○				SM0																																																																					
Word devices	<table><thead><tr><th rowspan="2">Devices</th><th rowspan="2">Symbol</th><th colspan="2">Accessibility</th><th colspan="3">Usable tasks</th><th rowspan="2">Description example</th></tr><tr><th>Read</th><th>Write</th><th>Normal</th><th>Event</th><th>NMI</th></tr></thead><tbody><tr><td>Data register</td><td>D</td><td>○</td><td>○</td><td rowspan="6">○</td><td rowspan="6">○</td><td rowspan="6">○</td><td>D0L</td></tr><tr><td>Link register</td><td>W</td><td>○</td><td>○</td><td>W1F : F</td></tr><tr><td>Multiple CPU area device</td><td>U□\G□</td><td>○</td><td>○</td><td>U3E0\G10100</td></tr><tr><td>Special register</td><td>SD</td><td>○</td><td>○</td><td>SD0</td></tr><tr><td>Motion register</td><td>#</td><td>○</td><td>○</td><td>#0F</td></tr><tr><td>Coasting timer</td><td>FT</td><td>○</td><td>×</td><td>FT</td></tr></tbody></table> <p>○ : usable × : unusable</p> <div>CAUTION</div> <p><Restrictions on write-enabled word devices></p> <p>1) Special register has predetermined applications in the system. Do not perform write to other than the user-set device.</p>							Devices	Symbol	Accessibility		Usable tasks			Description example	Read	Write	Normal	Event	NMI	Data register	D	○	○	○	○	○	D0L	Link register	W	○	○	W1F : F	Multiple CPU area device	U□\G□	○	○	U3E0\G10100	Special register	SD	○	○	SD0	Motion register	#	○	○	#0F	Coasting timer	FT	○	×	FT																								
Devices	Symbol	Accessibility		Usable tasks			Description example																																																																						
		Read	Write	Normal	Event	NMI																																																																							
Data register	D	○	○	○	○	○	D0L																																																																						
Link register	W	○	○				W1F : F																																																																						
Multiple CPU area device	U□\G□	○	○				U3E0\G10100																																																																						
Special register	SD	○	○				SD0																																																																						
Motion register	#	○	○				#0F																																																																						
Coasting timer	FT	○	×				FT																																																																						

1 OVERVIEW

Table of the operation control/transition control specification (continued)

Item	Specifications			Remark
Data type	(None)	16-bit integer type (signed)	-32768 to 32767	K10, D100, etc.
		16-bit integer type (unsigned)	0 to 65535	
	L	32-bit integer type (signed)	-2147483648 to 2147483647	2000000000, W100L, etc.
		32-bit integer type (unsigned)	0 to 4294967295	
	F	64-bit floating-point type (double precision real number type)	IEEE format	1.23, #10F, etc.
Constant	K	Decimal constant	The above data type symbol 'L' or '.' (decimal point)' provided at the end indicates the data type. The constant without the data type is regarded as the applicable minimum type.	K-100, H0FFL, etc. 'K' may be omitted.
	H	Hexadecimal constant		
Number of instructions	Binary operation		6	63 in total
	Bit operation		6	
	Sign		1	
	Standard function		15	
	Type conversion		6	
	Bit device status		2	
	Bit device control		5	
	Logical operation		4	
	Comparison operation		6	
	Motion dedicated function		2	
	Others		10	
Read/write response of input PX, output PY	Input response		Direct read control at instruction execution.	
	Output response		Direct write control at instruction execution.	

1 OVERVIEW

(2) Table of the operation control/transition instruction

Classification	Symbol	Function	Format	Basic steps	Usable step		Y/N transition's conditional expression	Section of reference
					F/FS	G		
Binary operation	=	Substitution	(D)=(S)	4	○	○	—	5.4.1
	+	Addition	(S1)+(S2)	4	○	○	—	5.4.2
	-	Subtraction	(S1)-(S2)	4	○	○	—	5.4.3
	*	Multiplication	(S1)*(S2)	4	○	○	—	5.4.4
	/	Division	(S1)/(S2)	4	○	○	—	5.4.5
	%	Remainder	(S1)%(S2)	4	○	○	—	5.4.6
Bit operation	~	Bit inversion (complement)	~(S)	2	○	○	—	5.5.1
	&	Bit logical AND	(S1)&(S2)	4	○	○	—	5.5.2
		Bit logical OR	(S1) (S2)	4	○	○	—	5.5.3
	^	Bit exclusive logical OR	(S1)^(S2)	4	○	○	—	5.5.4
	>>	Bit right shift	(S1)>>(S2)	4	○	○	—	5.5.5
	<<	Bit left shift	(S1)<<(S2)	4	○	○	—	5.5.6
Sign	-	Sign inversion (complement of 2)	-(S)	2	○	○	—	5.5.7
Standard function	SIN	Sine	SIN(S)	2	○	○	—	5.6.1
	COS	Cosine	COS(S)	2	○	○	—	5.6.2
	TAN	Tangent	TAN(S)	2	○	○	—	5.6.3
	ASIN	Arcsine	ASIN(S)	2	○	○	—	5.6.4
	ACOS	Arccosine	ACOS(S)	2	○	○	—	5.6.5
	ATAN	Arctangent	ATAN(S)	2	○	○	—	5.6.6
	SQRT	Square root	SQRT(S)	2	○	○	—	5.6.7
	LN	Natural logarithm	LN(S)	2	○	○	—	5.6.8
	EXP	Exponential operation	EXP(S)	2	○	○	—	5.6.9
	ABS	Absolute value	ABS(S)	2	○	○	—	5.6.10
	RND	Round-off	RND(S)	2	○	○	—	5.6.11
	FIX	Round-down	FIX(S)	2	○	○	—	5.6.12
	FUP	Round-up	FUP(S)	2	○	○	—	5.6.13
	BIN	BCD → BIN conversion	BIN(S)	2	○	○	—	5.6.14
	BCD	BIN → BCD conversion	BCD(S)	2	○	○	—	5.6.15
Type conversion	SHORT	Convert into 16-bit integer type (signed)	SHORT(S)	2	○	○	—	5.7.1
	USHORT	Convert into 16-bit integer type (unsigned)	USHORT(S)	2	○	○	—	5.7.2
	LONG	Convert into 32-bit integer type (signed)	LONG(S)	2	○	○	—	5.7.3
	ULONG	Convert into 32-bit integer type (unsigned)	ULONG(S)	2	○	○	—	5.7.4
	FLOAT	Regard as signed data and convert into 64-bit floating point type	FLOAT(S)	2	○	○	—	5.7.5
	UFLOAT	Regard as unsigned data and convert into 64-bit floating point type	UFLOAT(S)	2	○	○	—	5.7.6
Bit device status	(None)	ON (normally open contact)	(S)	2	○	○	○	5.8.1
	!	OFF (normally closed contact)	!(S)	2	○	○	○	5.8.2
Bit device control	SET	Device set	SET(D)	3	○	○	—	5.9.1
			SET(D)=(conditional expression)	4	○	○	—	
	RST	Device reset	RST(D)	3	○	○	—	5.9.2
			RST(D)=(conditional expression)	4	○	○	—	
	DOUT	Device output	DOUT(D),(S)	4	○	○	—	5.9.3
	DIN	Device input	DIN(D),(S)	4	○	○	—	5.9.4
	OUT	Bit device output	OUT(D)=(conditional expression)	4	○	○	—	5.9.5

1 OVERVIEW

Table of the operation control/transition instruction (continued)

Classification	Symbol	Function	Format	Basic steps	Usable step		Y/N transition's conditional expression	Section of reference
					F/FS	G		
Logical operation	(None)	Logical acknowledgment	(Conditional expression)	0	○	○	○	5.10.1
	!	Logical negation	!(Conditional expression)	2	○	○	○	5.10.2
	*	Logical AND	(Conditional expression) * (conditional expression)	4	○	○	○	5.10.3
	+	Logical OR	(Conditional expression) + (conditional expression)	4	○	○	○	5.10.4
Comparison operation	==	Equal to	(Conditional expression) == (conditional expression)	4	○	○	○	5.11.1
	!=	Not equal to	(Conditional expression) != (conditional expression)	4	○	○	○	5.11.2
	<	Less than	(Conditional expression) < (conditional expression)	4	○	○	○	5.11.3
	<=	Less than or equal to	(Conditional expression) <= (conditional expression)	4	○	○	○	5.11.4
	>	More than	(Conditional expression) > (conditional expression)	4	○	○	○	5.11.5
	>=	More than or equal to	(Conditional expression) >= (conditional expression)	4	○	○	○	5.11.6
Motion dedicated function	CHGV	Speed change request	CHGV((S1),(S2))	4	○	○	—	5.12.1
	CHGT	Torque limit value change request	CHGT((S1),(S2))	4	○	○	—	5.12.2
Others	EI	Event task enable	EI	1	○	○	—	5.13.1
	DI	Event task disable	DI	1	○	○	—	5.13.2
	NOP	No operation	NOP	1	○	○	—	5.13.3
	BMOV	Block transfer	BMOV(D),(S),(n)	6	○	○	—	5.13.4
	FMOV	Same data block transfer	FMOV(D),(S),(n)	6	○	○	—	5.13.5
	MULTW	Write device data to CPU shared memory of the self CPU	MULTW(D),(S),(n),(D1)	8	○	○	—	5.13.6
	MULTR	Read device data from CPU shared memory of the other CPU	MULTR(D),(S1),(S2),(n)	7	○	○	—	5.13.7
	TO	Write device data to intelligent function module.	TO(D1),(D2),(S),(n)	7	○	○	—	5.13.8
	FROM	Read device data from intelligent function module.	FROM(D),(S1),(S2),(n)	7	○	○	—	5.13.9
	TIME	Time to wait	TIME(S)	7	—	○	—	5.13.10

(3) Rough calculation expression of single program for operation control/transition program

$$\begin{aligned}
 & 2 + (1 + \text{Total number of basic steps in 1 block} \\
 & + \text{Number of 32-bit constants/1 block} \times 1 \\
 & + \text{Number of 64-bit constants/1 block} \times 3) \times \text{Number of blocks (steps)} \\
 & \quad (1 \text{ step} = 2 \text{ bytes})
 \end{aligned}$$

1.2.4 Positioning dedicated devices

(1) Positioning dedicated devices

The following section describes the positioning dedicated devices.

A range of up to 32 axes is valid in Q173DCPU, and a range of up to 8 axes is valid in Q172DCPU.

Refer to the "Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)", "Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for details of the positioning dedicated devices.

(a) Table of the internal relays

• Overall configuration

SV13		SV22	
Device No.	Purpose	Device No.	Purpose
M0 to	User device (2000 points)	M0 to	User device (2000 points)
M2000 to	Common device (320 points)	M2000 to	Common device (320 points)
M2320 to	Unusable (80 points)	M2320 to	Unusable (80 points)
M2400 to	Axis status (20 points × 32 axes)	M2400 to	Axis status (20 points × 32 axes) Real mode.....Each axis Virtual mode....Output module
M3040 to	Unusable (32 points)	M3040 to	Unusable (32 points)
M3072 to	Common device (Command signal) (64 points)	M3072 to	Common device (Command signal) (64 points)
M3136 to	Unusable (64 points)	M3136 to	Unusable (64 points)
M3200 to	Axis command signal (20 points × 32 axes)	M3200 to	Axis command signal (20 points × 32 axes) Real mode.....Each axis Virtual mode....Output module
M3840 to	User device (960 points)	M3840 to	Unusable (160 points)
		M4000 to	Virtual servomotor axis status (Note-1,2) (20 points × 32 axes) (Mechanical system setting axis only)
		M4640 to	Synchronous encoder axis status(Note-2) (4 points × 12 axes)
		M4688 to	Unusable (Note-1) (112 points)
M4799		M4799	

1 OVERVIEW

- Overall configuration (Continued)

SV13		SV22	
Device No.	Purpose	Device No.	Purpose
M4800 to M8191	User device (3392 points)	M4800 to	Virtual servomotor axis command signal (Note-1,2) (20 points × 32 axes) (Mechanical system setting axis only)
		M5440 to	Synchronous encoder axis command signal (Note-2) (4 points × 12 axes)
		M5488 to M8191	User device (Note-3) (2704 points)

(Note-1) : It can be used as an user device in the SV22 real mode only.

(Note-2) : Do not set the M4000 to M5487 as a latch range in the virtual mode.

(Note-3) : The cam axis command signal and smoothing clutch complete signal can be set as the optional device at the parameter.

1) Table of the axis statuses (SV13/SV22)

Device No.	Signal name	Device No.	Signal name
M2400 to M2419	Axis 1 status	M2720 to M2739	Axis 17 status
M2420 to M2439	Axis 2 status	M2740 to M2759	Axis 18 status
M2440 to M2459	Axis 3 status	M2760 to M2779	Axis 19 status
M2460 to M2479	Axis 4 status	M2780 to M2799	Axis 20 status
M2480 to M2499	Axis 5 status	M2800 to M2819	Axis 21 status
M2500 to M2519	Axis 6 status	M2820 to M2839	Axis 22 status
M2520 to M2539	Axis 7 status	M2840 to M2859	Axis 23 status
M2540 to M2559	Axis 8 status	M2860 to M2879	Axis 24 status
M2560 to M2579	Axis 9 status	M2880 to M2899	Axis 25 status
M2580 to M2599	Axis 10 status	M2900 to M2919	Axis 26 status
M2600 to M2619	Axis 11 status	M2920 to M2939	Axis 27 status
M2620 to M2639	Axis 12 status	M2940 to M2959	Axis 28 status
M2640 to M2659	Axis 13 status	M2960 to M2979	Axis 29 status
M2660 to M2679	Axis 14 status	M2980 to M2999	Axis 30 status
M2680 to M2699	Axis 15 status	M3000 to M3019	Axis 31 status
M2700 to M2719	Axis 16 status	M3020 to M3039	Axis 32 status

1 OVERVIEW

- Details of each axis

Device No.	Signal name	
M2400 + 20n	Positioning start complete	
M2401 + 20n	Positioning complete	
M2402 + 20n	In-position	
M2403 + 20n	Command in-position	
M2404 + 20n	Speed controlling	
M2405 + 20n	Speed/position switching latch signal	
M2406 + 20n	Zero pass signal	
M2407 + 20n	Error detection signal	
M2408 + 20n	Servo error detection signal	
M2409 + 20n	Home position return request signal	
M2410 + 20n	Home position return completion signal	
M2411 + 20n	External signals	FLS signal
M2412 + 20n		RLS signal
M2413 + 20n		STOP signal
M2414 + 20n		DOG/CHANGE signal
M2415 + 20n	Servo ready signal	
M2416 + 20n	Torque limiting signal	
M2417 + 20n	Unusable	
M2418 + 20n	Virtual mode continuation operation disable warning signal (SV22)	
M2419 + 20n	M-code outputting signal	

POINT

- (1) "n" in the above device No. shows the numerical value which correspond to axis No..
 Q173DCPU : Axis No.1 to No.32 (n=0 to 31)
 Q172DCPU : Axis No.1 to No.8 (n=0 to 7)
- (2) The device area more than 9 axes as an user device in the Q172DCPU.
 However, when the project of Q172DCPU is replaced with Q173DCPU, this area cannot be used.

2) Table of the axis command signals (SV13/SV22)

Device No.	Signal name	Device No.	Signal name
M3200 to M3219	Axis 1 command signal	M3520 to M3539	Axis 17 command signal
M3220 to M3239	Axis 2 command signal	M3540 to M3559	Axis 18 command signal
M3240 to M3259	Axis 3 command signal	M3560 to M3579	Axis 19 command signal
M3260 to M3279	Axis 4 command signal	M3580 to M3599	Axis 20 command signal
M3280 to M3299	Axis 5 command signal	M3600 to M3619	Axis 21 command signal
M3300 to M3319	Axis 6 command signal	M3620 to M3639	Axis 22 command signal
M3320 to M3339	Axis 7 command signal	M3640 to M3659	Axis 23 command signal
M3340 to M3359	Axis 8 command signal	M3660 to M3679	Axis 24 command signal
M3360 to M3379	Axis 9 command signal	M3680 to M3699	Axis 25 command signal
M3380 to M3399	Axis 10 command signal	M3700 to M3719	Axis 26 command signal
M3400 to M3419	Axis 11 command signal	M3720 to M3739	Axis 27 command signal
M3420 to M3439	Axis 12 command signal	M3740 to M3759	Axis 28 command signal
M3440 to M3459	Axis 13 command signal	M3760 to M3779	Axis 29 command signal
M3460 to M3479	Axis 14 command signal	M3780 to M3799	Axis 30 command signal
M3480 to M3499	Axis 15 command signal	M3800 to M3819	Axis 31 command signal
M3500 to M3519	Axis 16 command signal	M3820 to M3839	Axis 32 command signal

1 OVERVIEW

- Details of each axis

Device No.	SV13	SV22
M3200 + 20n	Stop command	Stop command
M3201 + 20n	Rapid stop command	Rapid stop command
M3202 + 20n	Forward rotation JOG start command	Forward rotation JOG start command
M3203 + 20n	Reverse rotation JOG start command	Reverse rotation JOG start command
M3204 + 20n	Complete signal OFF command	Complete signal OFF command
M3205 + 20n	Speed/position switching enable command	Speed/position switching enable command
M3206 + 20n	Unusable	Unusable
M3207 + 20n	Error reset command	Error reset command
M3208 + 20n	Servo error reset command	Servo error reset command
M3209 + 20n	External stop input disable at start command	External stop input disable at start command
M3210 + 20n	Unusable	Unusable
M3211 + 20n		
M3212 + 20n	Feed current value update request command	Feed current value update request command
M3213 + 20n	Unusable	Address clutch reference setting command
M3214 + 20n		Cam reference position setting command
M3215 + 20n	Servo OFF command	Servo OFF command
M3216 + 20n	Gain changing command	Gain changing command
M3217 + 20n	Unusable	Unusable
M3218 + 20n	Control loop changing command	Control loop changing command
M3219 + 20n	FIN signal	FIN signal

POINT

- "n" in the above device No. shows the numerical value which correspond to axis No..
 Q173DCPU : Axis No.1 to No.32 (n=0 to 31)
 Q172DCPU : Axis No.1 to No.8 (n=0 to 7)
- The device area more than 9 axes as an user device in the Q172DCPU.
 However, when the project of Q172DCPU is replaced with Q173DCPU, this area cannot be used.

3) Table of the virtual servomotor axis statuses (SV22 only)

Device No.	Signal name	Device No.	Signal name
M4000 to M4019	Axis 1 status	M4320 to M4339	Axis 17 status
M4020 to M4039	Axis 2 status	M4340 to M4359	Axis 18 status
M4040 to M4059	Axis 3 status	M4360 to M4379	Axis 19 status
M4060 to M4079	Axis 4 status	M4380 to M4399	Axis 20 status
M4080 to M4099	Axis 5 status	M4400 to M4419	Axis 21 status
M4100 to M4119	Axis 6 status	M4420 to M4439	Axis 22 status
M4120 to M4139	Axis 7 status	M4440 to M4459	Axis 23 status
M4140 to M4159	Axis 8 status	M4460 to M4479	Axis 24 status
M4160 to M4179	Axis 9 status	M4480 to M4499	Axis 25 status
M4180 to M4199	Axis 10 status	M4500 to M4519	Axis 26 status
M4200 to M4219	Axis 11 status	M4520 to M4539	Axis 27 status
M4220 to M4239	Axis 12 status	M4540 to M4559	Axis 28 status
M4240 to M4259	Axis 13 status	M4560 to M4579	Axis 29 status
M4260 to M4279	Axis 14 status	M4580 to M4599	Axis 30 status
M4280 to M4299	Axis 15 status	M4600 to M4619	Axis 31 status
M4300 to M4319	Axis 16 status	M4620 to M4639	Axis 32 status

1 OVERVIEW

- Details of each axis

Device No.	Signal name
M4000 + 20n	Positioning start complete
M4001 + 20n	Positioning complete
M4002 + 20n	Unusable
M4003 + 20n	Command in-position
M4004 + 20n	Speed controlling
M4005 + 20n	Unusable
M4006 + 20n	
M4007 + 20n	Error detection
M4008 + 20n	Unusable
M4009 + 20n	
M4010 + 20n	
M4011 + 20n	
M4012 + 20n	
M4013 + 20n	
M4014 + 20n	
M4015 + 20n	
M4016 + 20n	
M4017 + 20n	
M4018 + 20n	
M4019 + 20n	M-code outputting signal

POINT

- (1) "n" in the above device No. shows the numerical value which correspond to axis No..
Q173DCPU : Axis No.1 to No.32 (n=0 to 31)
Q172DCPU : Axis No.1 to No.8 (n=0 to 7)
- (2) The unused axis areas in the mechanical system program can be used as an user device.

4) Table of the virtual servomotor axis command signals (SV22 only)

Device No.	Signal name	Device No.	Signal name
M4800 to M4819	Axis 1 command signal	M5120 to M5139	Axis 17 command signal
M4820 to M4839	Axis 2 command signal	M5140 to M5159	Axis 18 command signal
M4840 to M4859	Axis 3 command signal	M5160 to M5179	Axis 19 command signal
M4860 to M4879	Axis 4 command signal	M5180 to M5199	Axis 20 command signal
M4880 to M4899	Axis 5 command signal	M5200 to M5219	Axis 21 command signal
M4900 to M4919	Axis 6 command signal	M5220 to M5239	Axis 22 command signal
M4920 to M4939	Axis 7 command signal	M5240 to M5259	Axis 23 command signal
M4940 to M4959	Axis 8 command signal	M5260 to M5279	Axis 24 command signal
M4960 to M4979	Axis 9 command signal	M5280 to M5299	Axis 25 command signal
M4980 to M4999	Axis 10 command signal	M5300 to M5319	Axis 26 command signal
M5000 to M5019	Axis 11 command signal	M5320 to M5339	Axis 27 command signal
M5020 to M5039	Axis 12 command signal	M5340 to M5359	Axis 28 command signal
M5040 to M5059	Axis 13 command signal	M5360 to M5379	Axis 29 command signal
M5060 to M5079	Axis 14 command signal	M5380 to M5399	Axis 30 command signal
M5080 to M5099	Axis 15 command signal	M5400 to M5419	Axis 31 command signal
M5100 to M5119	Axis 16 command signal	M5420 to M5439	Axis 32 command signal

1 OVERVIEW

- Details of each axis

Device No.	Signal name
M4800 + 20n	Stop command
M4801 + 20n	Rapid stop command
M4802 + 20n	Forward rotation JOG start command
M4803 + 20n	Reverse rotation JOG start command
M4804 + 20n	Complete signal OFF command
M4805 + 20n	Unusable
M4806 + 20n	
M4807 + 20n	Error reset command
M4808 + 20n	Unusable
M4809 + 20n	External stop input disable at start command
M4810 + 20n	Unusable
M4811 + 20n	
M4812 + 20n	
M4813 + 20n	
M4814 + 20n	
M4815 + 20n	
M4816 + 20n	
M4817 + 20n	
M4818 + 20n	
M4819 + 20n	FIN signal

POINT
(1) "n" in the above device No. shows the numerical value which correspond to axis No.. Q173DCPU : Axis No.1 to No.32 (n=0 to 31) Q172DCPU : Axis No.1 to No.8 (n=0 to 7) (2) The unused axis areas in the mechanical system program can be used as an user device.

5) Table of the synchronous encoder axis statuses (SV22 only)

Device No.	Signal name	
M4640	Axis 1	Error detection
M4641		External signal TREN
M4642		Virtual mode continuation operation disable warning
M4643		Unusable
M4644	Axis 2	Error detection
M4645		External signal TREN
M4646		Virtual mode continuation operation disable warning
M4647		Unusable
M4648	Axis 3	Error detection
M4649		External signal TREN
M4650		Virtual mode continuation operation disable warning
M4651		Unusable
M4652	Axis 4	Error detection
M4653		External signal TREN
M4654		Virtual mode continuation operation disable warning
M4655		Unusable
M4656	Axis 5	Error detection
M4657		External signal TREN
M4658		Virtual mode continuation operation disable warning
M4659		Unusable
M4660	Axis 6	Error detection
M4661		External signal TREN
M4662		Virtual mode continuation operation disable warning
M4663		Unusable
M4664	Axis 7	Error detection
M4665		External signal TREN
M4666		Virtual mode continuation operation disable warning
M4667		Unusable
M4668	Axis 8	Error detection
M4669		External signal TREN
M4670		Virtual mode continuation operation disable warning
M4671		Unusable
M4672	Axis 9	Error detection
M4673		External signal TREN
M4674		Virtual mode continuation operation disable warning
M4675		Unusable
M4676	Axis 10	Error detection
M4677		External signal TREN
M4678		Virtual mode continuation operation disable warning
M4679		Unusable
M4680	Axis 11	Error detection
M4681		External signal TREN
M4682		Virtual mode continuation operation disable warning
M4683		Unusable
M4684	Axis 12	Error detection
M4685		External signal TREN
M4686		Virtual mode continuation operation disable warning
M4687		Unusable

POINT

- (1) The range of axis No.1 to 8 is valid in the Q172DCPU.
- (2) The device area more than 9 axes as an user device in the Q172DCPU.
However, when the project of Q172DCPU is replaced with Q173DCPU, this area cannot be used.

6) Table of the synchronous encoder axis command signals (SV22 only)

Device No.	Signal name	
M5440	Axis 1	Error reset
M5441		Unusable
M5442		Unusable
M5443		Unusable
M5444	Axis 2	Error reset
M5445		Unusable
M5446		Unusable
M5447		Unusable
M5448	Axis 3	Error reset
M5449		Unusable
M5450		Unusable
M5451		Unusable
M5452	Axis 4	Error reset
M5453		Unusable
M5454		Unusable
M5455		Unusable
M5456	Axis 5	Error reset
M5457		Unusable
M5458		Unusable
M5459		Unusable
M5460	Axis 6	Error reset
M5461		Unusable
M5462		Unusable
M5463		Unusable
M5464	Axis 7	Error reset
M5465		Unusable
M5466		Unusable
M5467		Unusable
M5468	Axis 8	Error reset
M5469		Unusable
M5470		Unusable
M5471		Unusable
M5472	Axis 9	Error reset
M5473		Unusable
M5474		Unusable
M5475		Unusable
M5476	Axis 10	Error reset
M5477		Unusable
M5478		Unusable
M5479		Unusable
M5480	Axis 11	Error reset
M5481		Unusable
M5482		Unusable
M5483		Unusable
M5484	Axis 12	Error reset
M5485		Unusable
M5486		Unusable
M5487		Unusable

POINT

- (1) The range of axis No.1 to 8 is valid in the Q172DCPU.
- (2) The device area more than 9 axes as an user device in the Q172DCPU.
However, when the project of Q172DCPU is replaced with Q173DCPU, this area cannot be used.

1 OVERVIEW

7) Table of the common devices (SV13/SV22)

SV13			SV22			Refresh cycle	Fetch cycle	Signal direction	Remark (Note-4)	
Device No.	Signal name		Device No.	Signal name						
M2000	PLC ready flag		M2000	PLC ready flag			Main cycle	Command signal	M3072	
M2001	Axis1	Start accept flag (32 points)	M2001	Axis1	Start accept flag (32 points)	Operation cycle		Status signal (Note-1,2)		
to	to		to	to						
M2032	Axis32		M2032	Axis32						
M2033	Unusable (2 points)		M2033	Unusable (2 points)		—	—	—		
M2034			M2034							
M2035	Motion error history clear request flag		M2035	Motion error history clear request flag			Main cycle	Command signal	M3080	
M2036	Unusable (2 points)		M2036	Unusable (2 points)		—	—	—		
M2037			M2037							
M2038	Motion SFC debugging flag		M2038	Motion SFC debugging flag		At debug mode transition		Status signal		
M2039	Motion error detection flag		M2039	Motion error detection flag						Immedi-ate
M2040	Speed switching point specified flag		M2040	Speed switching point specified flag			At start	Command signal	M3073	
M2041	System setting error flag		M2041	System setting error flag		Operation cycle		Status signal		
M2042	All axes servo ON command		M2042	All axes servo ON command				Operation cycle	Command signal	M3074
M2043	Unusable (4 points)		M2043	Real mode /virtual mode switching request			At virtual mode transition			M3075
M2044			Real mode/virtual mode switching status							
M2045			Real mode/virtual mode switching error detection flag		At virtual mode transition		Status signal			
M2046			Out-of-sync warning							
M2047	Motion slot fault detection flag		M2047	Motion slot fault detection flag		Operation cycle		Main cycle	Command signal	M3076
M2048	JOG operation simultaneous start command		M2048	JOG operation simultaneous start command						
M2049	All axes servo ON accept flag		M2049	All axes servo ON accept flag		Operation cycle		Status signal		
M2050	Unusable		M2050	Unusable		—	—	—		
M2051	Manual pulse generator 1 enable flag		M2051	Manual pulse generator 1 enable flag			Main cycle	Command signal	M3077	
M2052	Manual pulse generator 2 enable flag		M2052	Manual pulse generator 2 enable flag					M3078	
M2053	Manual pulse generator 3 enable flag		M2053	Manual pulse generator 3 enable flag					M3079	

1 OVERVIEW

Table of the common devices (SV13/SV22) (continued)

SV13			SV22			Refresh cycle	Fetch cycle	Signal direction	Remark (Note-4)	
Device No.	Signal name		Device No.	Signal name						
M2054	Operation cycle over flag		M2054	Operation cycle over flag		Operation cycle		Status signal		
M2055	Unusable (6 points)		M2055	Unusable (6 points)		—	—	—		
to			to							
M2060			M2060							
M2061	Axis 1	Speed change accepting flag (32 axes)	M2061	Axis 1	Speed change accepting flag (32 axes)	Operation cycle		Status signal (Note-1,2)		
to	to		to	to						
M2092	Axis 32		M2092	Axis 32						
M2093	Unusable (35 points)		M2093	Unusable (8 points)		—	—	—		
to			to							
M2100										
M2101			Axis 1	Synchronous encoder current value changing flag (Note-3) (12 axes)	Operation cycle		Status signal (Note-1,2)			
to			to							
M2112			Axis 12							
M2113			Unusable (15 points)		—	—	—			
to								to		
M2127			M2127							
M2128	Axis 1	Automatic decelerating flag (32 axes)	M2128	Axis 1	Automatic decelerating flag (32 axes)	Operation cycle		Status signal (Note-1,2)		
to	to		to	to						
M2159	Axis 32	M2159	Axis 32							
M2160	Unusable (80 points)		M2160	Unusable (Note-5) (64 points)		—	—	—		
to			to							
M2223										
M2224			Unusable (16 points)		to					
to										
M2239			M2239							
M2240	Axis 1	Speed change "0" accepting flag (32 axes)	M2240	Axis 1	Speed change "0" accepting flag (32 axes)	Operation cycle		Status signal (Note-1,2)		
to	to		to	to						
M2271	Axis 32	M2271	Axis 32							
M2272	Axis 1	Control loop monitor status (32 axes)	M2272	Axis 1	Control loop monitor status (32 axes)					
to	to		to	to						
M2303	Axis 32	M2303	Axis 32							
M2304	Unusable (16 points)		M2304	Unusable (16 points)		—	—	—		
to			to							
M2319			M2319							

(Note-1) : The range of axis No.1 to 8 is valid in the Q172DCPU.

(Note-2) : Device area of 9 axes or more is unusable in the Q172DCPU.

(Note-3) : It is unusable in the SV13/SV22 real mode.

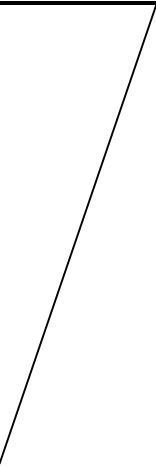
(Note-4) : It can also be ordered the device of a remark column.

(Note-5) : These devices can be used as the clutch statuses.

The clutch status can also be set as the optional device at the clutch parameter.

Refer to Section 7.2.2 of the "Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for details.

8) Table of the common devices (Command signal) (SV13/SV22)

Device No.	Signal name	Refresh cycle	Fetch cycle	Signal direction	Remark (Note-1) , (Note-2)
M3072	PLC ready flag		Main cycle	Command signal	M2000
M3073	Speed switching point specified flag		At start		M2040
M3074	All axes servo ON command		Operation cycle		M2042
M3075	Real mode/virtual mode change request (SV22)		At virtual mode transition		M2043
M3076	JOG operation simultaneous start command		Main cycle		M2048
M3077	Manual pulse generator 1 enable flag				M2051
M3078	Manual pulse generator 2 enable flag				M2052
M3079	Manual pulse generator 3 enable flag				M2053
M3080	Motion error history clear request flag				M2035
M3081 to M3135	Unusable ^(Note-3) (55 points)		—		—

(Note-1) : The state of a device is not in agreement when the device of a remark column is turned ON/OFF directly. In addition, when the request from a data register and the request from the above device are performed simultaneously, the request from the above device becomes effective.

(Note-2) : It can also be ordered the device of a remark column.

(Note-3): Do not use it as an user device. It can be used as a device that performs automatic refresh because of area for the reserve of command signal.

POINT

The device of a remark column turns ON by OFF to ON of the above device, and turns OFF by ON to OFF of the above device.
The command signal cannot be turned ON/OFF by the PLC CPU in the automatic refresh because the statuses and commands are mixed together in M2000 to M2053. Use the above devices in the case.
And, it can also be turned ON/OFF by the data register.
Refer to Section 3.2.3 of the "Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" or Section 4.2.8 of the "Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for details.

(b) Table of the data registers

- Overall configuration

SV13		SV22	
Device No.	Application	Device No.	Application
D0	Axis monitor device (20 points × 32 axes)	D0	Axis monitor device (20 points × 32 axes)
to		to	Real mode.....Each axis Virtual mode....Output module
D640	Control change register (2 points × 32 axes)	D640	Control change register
to		to	(2 points × 32 axes)
D704	Common device (Command signal) (54 points)	D704	Common device (Command signal)
to		to	(54 points)
D758	Unusable (42 points)	D758	Unusable
to		to	(42 points)
D800	User device (7392 points)	D800	Virtual servomotor axis monitor device (Note)
to		to	(10 points × 32 axes) (Mechanical system setting axis only)
		D1120	Synchronous encoder axis monitor device (Note) (10 points × 12 axes)
		to	
		D1240	Cam axis monitor device (Note)
to		to	(10 points × 32 axes)
to		D1560	User device (6632 points)
		to	
	to		
D8191		D8191	

(Note) : It can be used as an user device in the SV22 real mode only.

1) Table of the each axis monitor devices (SV13/SV22)

Device No.	Signal name	Device No.	Signal name
D0 to D19	Axis 1 monitor device	D320 to D339	Axis 17 monitor device
D20 to D39	Axis 2 monitor device	D340 to D359	Axis 18 monitor device
D40 to D59	Axis 3 monitor device	D360 to D379	Axis 19 monitor device
D60 to D79	Axis 4 monitor device	D380 to D399	Axis 20 monitor device
D80 to D99	Axis 5 monitor device	D400 to D419	Axis 21 monitor device
D100 to D119	Axis 6 monitor device	D420 to D439	Axis 22 monitor device
D120 to D139	Axis 7 monitor device	D440 to D459	Axis 23 monitor device
D140 to D159	Axis 8 monitor device	D460 to D479	Axis 24 monitor device
D160 to D179	Axis 9 monitor device	D480 to D499	Axis 25 monitor device
D180 to D199	Axis 10 monitor device	D500 to D519	Axis 26 monitor device
D200 to D219	Axis 11 monitor device	D520 to D539	Axis 27 monitor device
D220 to D239	Axis 12 monitor device	D540 to D559	Axis 28 monitor device
D240 to D259	Axis 13 monitor device	D560 to D579	Axis 29 monitor device
D260 to D279	Axis 14 monitor device	D580 to D599	Axis 30 monitor device
D280 to D299	Axis 15 monitor device	D600 to D619	Axis 31 monitor device
D300 to D319	Axis 16 monitor device	D620 to D639	Axis 32 monitor device

1 OVERVIEW

- Details of each axis

Device No.	SV13/SV22(Real mode)	SV22(Virtual mode)	Signal direction
D0 + 20n D1 + 20n	Feed current value	Feed current value/roller cycle speed	Monitor device
D2 + 20n D3 + 20n	Real current value	Real current value	
D4 + 20n D5 + 20n	Deviation counter value	Deviation counter value	
D6 + 20n	Minor error code	Minor error code	
D7 + 20n	Major error code	Major error code	
D8 + 20n	Servo error code	Servo error code	
D9 + 20n	Home position return re-travel value	Hold	
D10 + 20n D11 + 20n	Travel value after proximity dog ON	Hold	
D12 + 20n	Execute program No.	—	
D13 + 20n	M-code	—	
D14 + 20n	Torque limit value	Torque limit value	
D15 + 20n	Data set pointer for constant-speed control	—	
D16 + 20n D17 + 20n	Unusable (Note-1)	Unusable (Note-1)	—
D18 + 20n D19 + 20n	Real current value at stop input	Hold	Monitor device

(Note-1): It can be used as the travel value change register.

The travel value change register can be set to the device optionally in the servo program.

Refer to Section 6.15 of the "Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for details.

POINT
<p>(1) "n" in the above device No. shows the numerical value which correspond to axis No..</p> <p>Q173DCPU : Axis No.1 to No.32 (n=0 to 31)</p> <p>Q172DCPU : Axis No.1 to No.8 (n=0 to 7)</p> <p>(2) The device area more than 9 axes as an user device in the Q172DCPU.</p> <p>However, when the project of Q172DCPU is replaced with Q173DCPU, this area cannot be used.</p>

2) Table of the control change registers (SV13/SV22)

Device No.	Signal name	Device No.	Signal name
D640 D641	Axis 1 JOG speed setting register	D672 D673	Axis 17 JOG speed setting register
D642 D643	Axis 2 JOG speed setting register	D674 D675	Axis 18 JOG speed setting register
D644 D645	Axis 3 JOG speed setting register	D676 D677	Axis 19 JOG speed setting register
D646 D647	Axis 4 JOG speed setting register	D678 D679	Axis 20 JOG speed setting register
D648 D649	Axis 5 JOG speed setting register	D680 D681	Axis 21 JOG speed setting register
D650 D651	Axis 6 JOG speed setting register	D682 D683	Axis 22 JOG speed setting register
D652 D653	Axis 7 JOG speed setting register	D684 D685	Axis 23 JOG speed setting register
D654 D655	Axis 8 JOG speed setting register	D686 D687	Axis 24 JOG speed setting register
D656 D657	Axis 9 JOG speed setting register	D688 D689	Axis 25 JOG speed setting register
D658 D659	Axis 10 JOG speed setting register	D690 D691	Axis 26 JOG speed setting register
D660 D661	Axis 11 JOG speed setting register	D692 D693	Axis 27 JOG speed setting register
D662 D663	Axis 12 JOG speed setting register	D694 D695	Axis 28 JOG speed setting register
D664 D665	Axis 13 JOG speed setting register	D696 D697	Axis 29 JOG speed setting register
D666 D667	Axis 14 JOG speed setting register	D698 D699	Axis 30 JOG speed setting register
D668 D669	Axis 15 JOG speed setting register	D700 D701	Axis 31 JOG speed setting register
D670 D671	Axis 16 JOG speed setting register	D702 D703	Axis 32 JOG speed setting register

POINT

- (1) The range of axis No.1 to 8 is valid in the Q172DCPU.
- (2) The device area more than 9 axes as an user device in the Q172DCPU.
However, when the project of Q172DCPU is replaced with Q173DCPU, this area cannot be used.

[illegible]

3) Table of the virtual servomotor axis monitor devices (SV22 only)

Device No.	Signal name	Device No.	Signal name
D800 to D809	Axis 1 monitor device	D960 to D969	Axis 17 monitor device
D810 to D819	Axis 2 monitor device	D970 to D979	Axis 18 monitor device
D820 to D829	Axis 3 monitor device	D980 to D989	Axis 19 monitor device
D830 to D839	Axis 4 monitor device	D990 to D999	Axis 20 monitor device
D840 to D849	Axis 5 monitor device	D1000 to D1009	Axis 21 monitor device
D850 to D859	Axis 6 monitor device	D1010 to D1019	Axis 22 monitor device
D860 to D869	Axis 7 monitor device	D1020 to D1029	Axis 23 monitor device
D870 to D879	Axis 8 monitor device	D1030 to D1039	Axis 24 monitor device
D880 to D889	Axis 9 monitor device	D1040 to D1049	Axis 25 monitor device
D890 to D899	Axis 10 monitor device	D1050 to D1059	Axis 26 monitor device
D900 to D909	Axis 11 monitor device	D1060 to D1069	Axis 27 monitor device
D910 to D919	Axis 12 monitor device	D1070 to D1079	Axis 28 monitor device
D920 to D929	Axis 13 monitor device	D1080 to D1089	Axis 29 monitor device
D930 to D939	Axis 14 monitor device	D1090 to D1099	Axis 30 monitor device
D940 to D949	Axis 15 monitor device	D1100 to D1109	Axis 31 monitor device
D950 to D959	Axis 16 monitor device	D1110 to D1119	Axis 32 monitor device

1 OVERVIEW

- Details of each axis

Device No.	Signal name
D800 + 10n D801 + 10n	Feed current value
D802 + 10n	Minor error code
D803 + 10n	Major error code
D804 + 10n	Execute program No.
D805 + 10n	M-code
D806 + 10n D807 + 10n	Current value after virtual servomotor axis main shaft's differential gear
D808 + 10n	Error search output axis No.
D809 + 10n	Data set pointer for constant-speed control

POINT

- | |
|---|
| <p>(1) "n" in the above device No. shows the numerical value which correspond to axis No..
Q173DCPU : Axis No.1 to No.32 (n=0 to 31)
Q172DCPU : Axis No.1 to No.8 (n=0 to 7)</p> <p>(2) The unused axis areas in the mechanical system program can be used as an user device.</p> |
|---|

4) Table of the synchronous encoder axis monitor devices (SV22 only)

Device No.	Signal name
D1120 to D1129	Axis 1 monitor device
D1130 to D1139	Axis 2 monitor device
D1140 to D1149	Axis 3 monitor device
D1150 to D1159	Axis 4 monitor device
D1160 to D1169	Axis 5 monitor device
D1170 to D1179	Axis 6 monitor device
D1180 to D1189	Axis 7 monitor device
D1190 to D1199	Axis 8 monitor device
D1200 to D1209	Axis 9 monitor device
D1210 to D1219	Axis 10 monitor device
D1220 to D1229	Axis 11 monitor device
D1230 to D1239	Axis 12 monitor device

1 OVERVIEW

- Details of each axis

Device No.	Signal name
D1120 + 10n D1121 + 10n	Current value
D1122 + 10n	Minor error code
D1123 + 10n	Major error code
D1124 + 10n D1125 + 10n	Unusable
D1126 + 10n D1127 + 10n	Current value after synchronous encoder axis main shaft's differential gear
D1128 + 10n	Error search output axis No.
D1129 + 10n	Unusable

POINT

- | |
|---|
| <p>(1) "n" in the above device No. shows the numerical value which correspond to axis No..</p> <p>Q173DCPU : Axis No.1 to No.32 (n=0 to 31)</p> <p>Q172DCPU : Axis No.1 to No.8 (n=0 to 7)</p> <p>(2) The device area more than 9 axes as an user device in the Q172DCPU.</p> <p>However, when the project of Q172DCPU is replaced with Q173DCPU, this area cannot be used.</p> |
|---|

5) Table of the cam axis monitor devices (SV22 only)

Device No.	Signal name	Device No.	Signal name
D1240 to D1249	Axis 1 monitor device	D1400 to D1409	Axis 17 monitor device
D1250 to D1259	Axis 2 monitor device	D1410 to D1419	Axis 18 monitor device
D1260 to D1269	Axis 3 monitor device	D1420 to D1429	Axis 19 monitor device
D1270 to D1279	Axis 4 monitor device	D1430 to D1439	Axis 20 monitor device
D1280 to D1289	Axis 5 monitor device	D1440 to D1449	Axis 21 monitor device
D1290 to D1299	Axis 6 monitor device	D1450 to D1459	Axis 22 monitor device
D1300 to D1309	Axis 7 monitor device	D1460 to D1469	Axis 23 monitor device
D1310 to D1319	Axis 8 monitor device	D1470 to D1479	Axis 24 monitor device
D1320 to D1329	Axis 9 monitor device	D1480 to D1489	Axis 25 monitor device
D1330 to D1339	Axis 10 monitor device	D1490 to D1499	Axis 26 monitor device
D1340 to D1349	Axis 11 monitor device	D1500 to D1509	Axis 27 monitor device
D1350 to D1359	Axis 12 monitor device	D1510 to D1519	Axis 28 monitor device
D1360 to D1369	Axis 13 monitor device	D1520 to D1529	Axis 29 monitor device
D1370 to D1379	Axis 14 monitor device	D1530 to D1539	Axis 30 monitor device
D1380 to D1389	Axis 15 monitor device	D1540 to D1549	Axis 31 monitor device
D1390 to D1399	Axis 16 monitor device	D1550 to D1559	Axis 32 monitor device

1 OVERVIEW

- Details of each axis

Device No.	Signal name
D1240 + 10n	Unusable
D1241 + 10n	Execute cam No.
D1242 + 10n	Execute stroke amount
D1243 + 10n	
D1244 + 10n	Current value within 1 cam shaft revolution
D1245 + 10n	
D1246 + 10n	Unusable
D1247 + 10n	
D1248 + 10n	
D1249 + 10n	

POINT

- (1) "n" in the above device No. shows the numerical value which correspond to axis No.
Q173DCPU : Axis No.1 to No.32 (n=0 to 31)
Q172DCPU : Axis No.1 to No.8 (n=0 to 7)
- (2) The unused axis areas in the mechanical system program can be used as an user device.

1 OVERVIEW

6) Table of the common devices (SV13/SV22)

Device No.	Signal name		Signal direction	Device No.	Signal name		Signal direction		
D704	PLC ready flag request		Command device	D740	Axis 21	Manual pulse generators 1 pulse input magnification setting register (Note-1,2)	Command device		
D705	Speed switching point specified flag request			D741	Axis 22				
D706	All axes servo ON command request			D742	Axis 23				
D707	Real mode/virtual mode switching request (SV22)			D743	Axis 24				
D708	JOG operation simultaneous start command request			D744	Axis 25				
D709	Unusable		—	D745	Axis 26			Manual pulse generator 1 smoothing magnification setting register	Command device
D710 to D713	JOG operation simultaneous start axis setting register		D746	Axis 27					
D714 D715	Manual pulse generator axis 1 No. setting register		D747	Axis 28					
			D748	Axis 29					
			D749	Axis 30					
D716 D717	Manual pulse generator axis 2 No. setting register		D750	Axis 32					
			D751	Axis 32					
			D752	Manual pulse generator 1 smoothing magnification setting register					
D718 D719	Manual pulse generator axis 3 No. setting register		D753	Manual pulse generator 2 smoothing magnification setting register					
			D754	Manual pulse generator 3 smoothing magnification setting register					
			D755	Manual pulse generator 1 enable flag request					
D720	Axis 1	Manual pulse generators 1 pulse input magnification setting register (Note-1,2)	D756	Manual pulse generator 2 enable flag request	Unusable (42 points)	—			
D721	Axis 2		D757	Manual pulse generator 3 enable flag request					
D722	Axis 3		to	D758					
D723	Axis 4								
D724	Axis 5								
D725	Axis 6								
D726	Axis 7								
D727	Axis 8								
D728	Axis 9								
D729	Axis 10								
D730	Axis 11								
D731	Axis 12								
D732	Axis 13								
D733	Axis 14								
D734	Axis 15								
D735	Axis 16								
D736	Axis 17								
D737	Axis 18								
D738	Axis 19								
D739	Axis 20		D799						

(Note-1) : The range of axis No.1 to 8 is valid in the Q172DCPU.

(Note-2) : Device area of 9 axes or more is unusable in the Q172DCPU.

2. STRUCTURE OF THE MOTION CPU PROGRAM

Motion CPU programs is created in the Motion SFC of flowchart format. The motion control of servomotors is performed using the real-mode servo programs specified by motion-control steps in a Motion SFC program in SV13/SV22 real mode. Virtual servomotors in a mechanical system program are controlled using the virtual mode servo programs specified by motion-control steps so as to enable synchronous control in SV22 virtual mode. Refer to the documents below for the details of Motion SFC programs, motion control in real mode, and motion control in virtual mode.

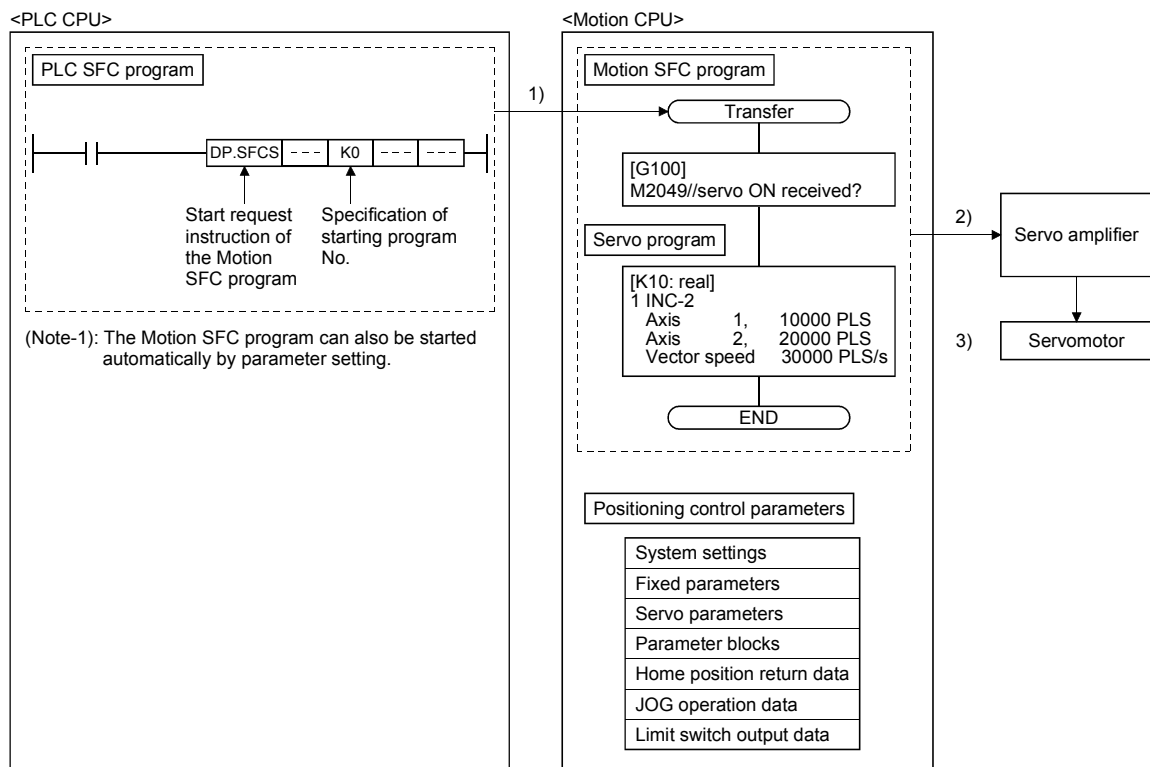
Item	Reference manual
Motion SFC program	Section 4 in this manual
Motion control in SV13/SV22 real mode (Servo program)	Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)
Motion control in SV22 virtual mode (Mechanical system program)	Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)

2 STRUCTURE OF THE MOTION CPU PROGRAM

2.1 Motion Control in SV13/SV22 Real Mode

- (1) System with servomotor is controlled directly using the servo program in (SV13/SV22) real mode.
- (2) Setting of the positioning parameter and creation of the servo program/ Motion SFC program are required.
- (3) The procedure of positioning control is shown below:
 - 1) Motion SFC program is requested to start using the D(P).SFCS instruction of the PLC program.
(Motion SFC program can also be started automatically by parameter setting.)
 - ↓
 - 2) Execute the positioning control using the specified the Motion SFC program.
(Output to the servo amplifier)
 - ↓
 - 3) The servomotor is controlled.

Program structure in SV13/SV22 real mode

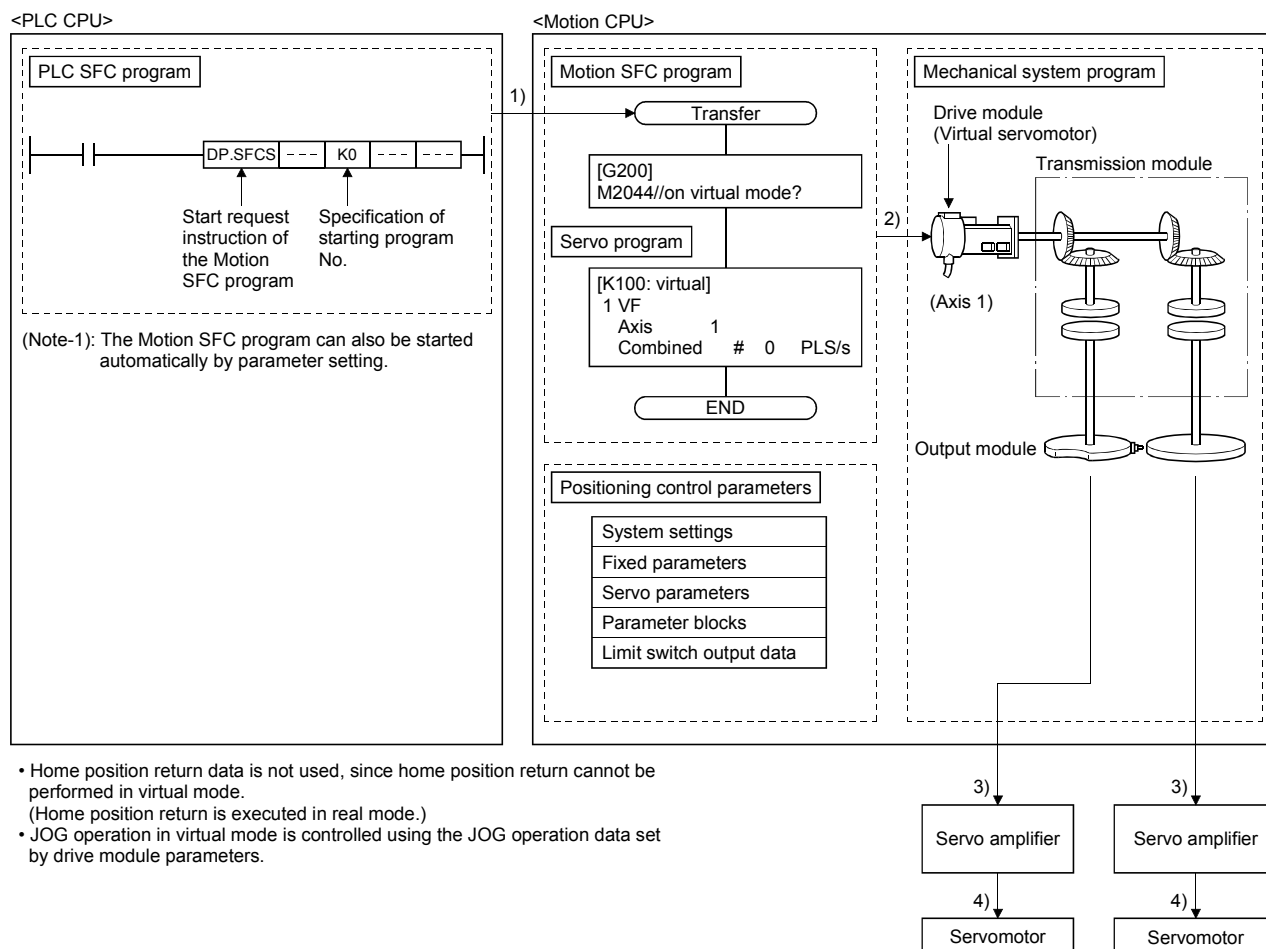


2 STRUCTURE OF THE MOTION CPU PROGRAM

2.2 Motion Control in SV22 Virtual Mode

- (1) Software-based synchronous control is performed using the mechanical system program constructed by virtual main shaft and mechanical module in (SV22) virtual mode.
- (2) Mechanical system programs is required in addition to the positioning parameter, servo program/Motion SFC program used in real mode.
- (3) The procedure of positioning control in virtual model is shown below:
 - 1) Motion SFC program for virtual mode is requested to start using the D(P).SFCS instruction of the PLC program.
(Motion SFC program can also be started automatically by parameter setting.)
 - ↓
 - 2) The virtual servomotor in the mechanical system program is started.
 - ↓
 - 3) Output the operation result obtained through the transmission module to the servo amplifier set as the output module.
 - ↓
 - 4) The servomotor is controlled.

Program structure in SV22 virtual mode



[illegible]

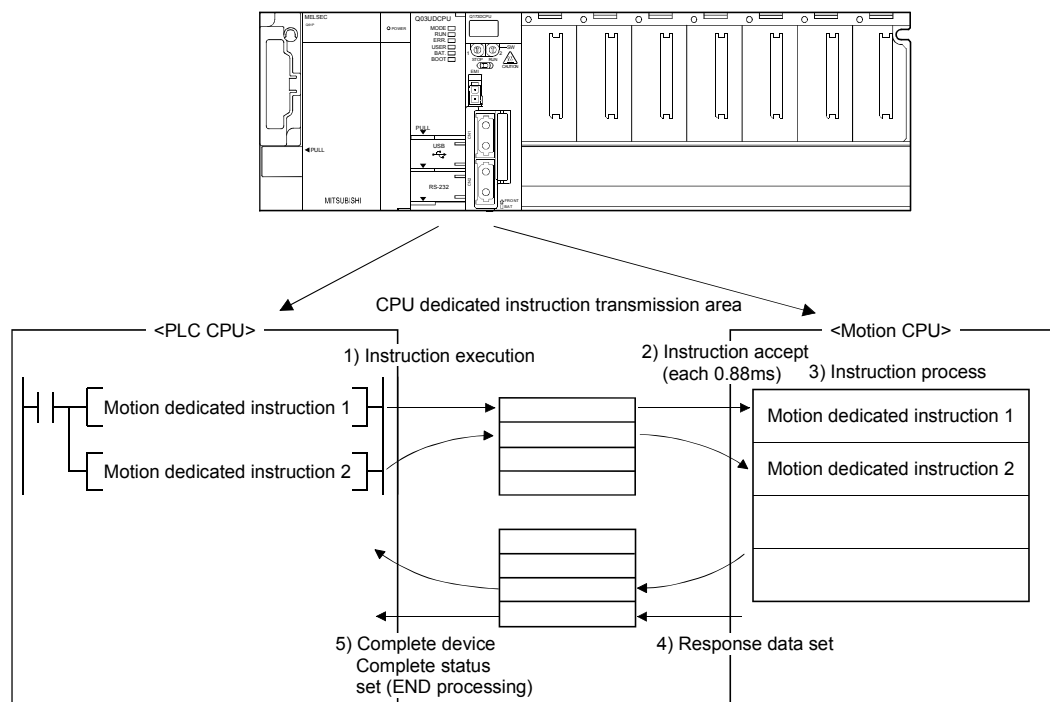
3. MOTION DEDICATED PLC INSTRUCTION

3.1 Outline of Motion Dedicated PLC Instruction

Motion dedicated PLC instruction is used to access the device data and start-up program of Motion CPU from PLC CPU.

Motion dedicated PLC instruction is transmitted through the CPU dedicated instruction transmission area set up in system area on the shared memory at the Multiple CPU high speed transmission.

Outline operation for Motion dedicated PLC instruction is shown below.



3.2 Motion Dedicated PLC Instruction

The Motion dedicated PLC instruction that can be executed toward the Motion CPU which installed the operating system software (SV13/SV22) for Q173DCPU/ Q172DCPU is shown below.

Instruction	Description
D(P).SFCS	Start request of the specified Motion SFC program
D(P).SVST	Start request of the specified servo program
D(P).CHGA	Current value change request of the specified axis
D(P).CHGV	Speed change request of the specified axis
D(P).CHGT	Torque control value change request of the specified axis
D(P).DDWR	Write device data of the self CPU to the device of other CPU
D(P).DDRD	Read device data of other CPU to the device of self CPU
D(P).GINT	Execute request of an event task of Motion SFC program

3 MOTION DEDICATED PLC INSTRUCTION

3.2.1 Motion SFC start request from the PLC CPU to the Motion CPU:D(P).SFCS (PLC instruction: D(P).SFCS)

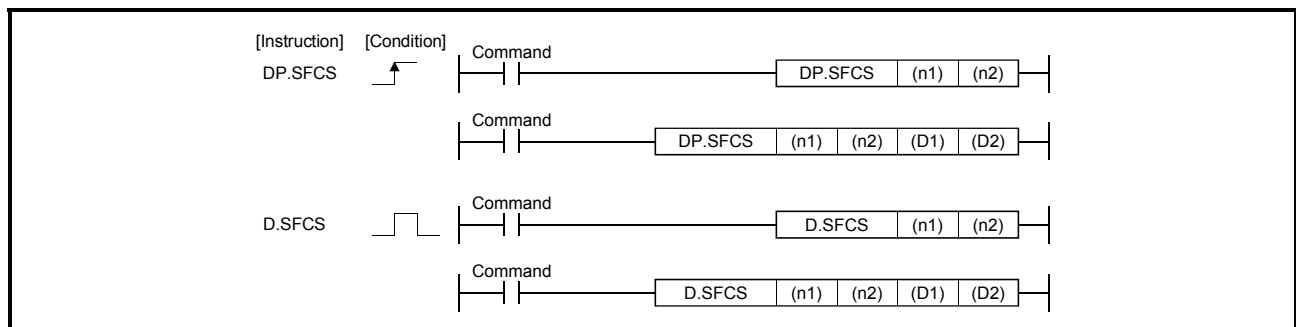
Setting data (Note-3)	Usable devices															
	Internal devices (System, User)				File register		Indirect specified	Link direct device J□\G			Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word			Bit	Word		Bit	Word		Bit	Word		Decimal, Hexadecimal K, H	Real character string	
		Digit specified A	Digit specified B	Digit specified												
(n1)		○	○	○		○	○							○		
(n2)		○	○	○		○	○							○		
(D1) (Note-1)	△ (Note-2)				△ (Note-2)											
(D2) (Note-1)		△ (Note-2)				△ (Note-2)	○									

○ : Usable △ : Usable partly

(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : Local devices cannot be used.

(Note-3) : Setting data (n1) to (D2) : Index qualification possible



[Setting data]

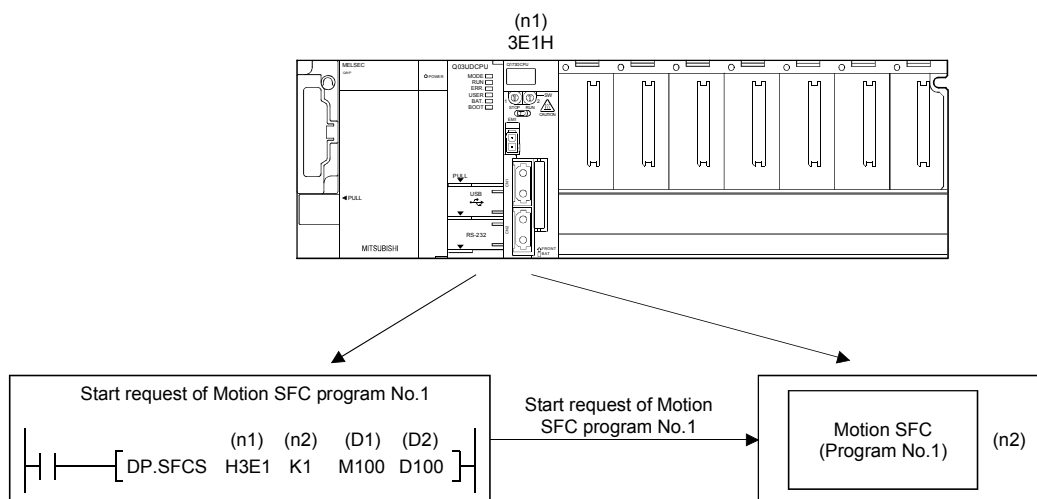
Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(n2)	Motion SFC program No. to start.	User	16-bit binary
(D1) ^(Note-1)	Complete devices (D1+0) : Device which make turn on for one scan at accept completion of instruction. (D1+1) : Device which make turn on for one scan at accept abnormal completion of instruction. (“D1+0” also turns on at the abnormal completion.)	System	Bit
(D2) ^(Note-1)	Complete status storage device	System	16-bit binary

(Note-1) : Omission possible with both of (D1) and (D2) omission.

3 MOTION DEDICATED PLC INSTRUCTION

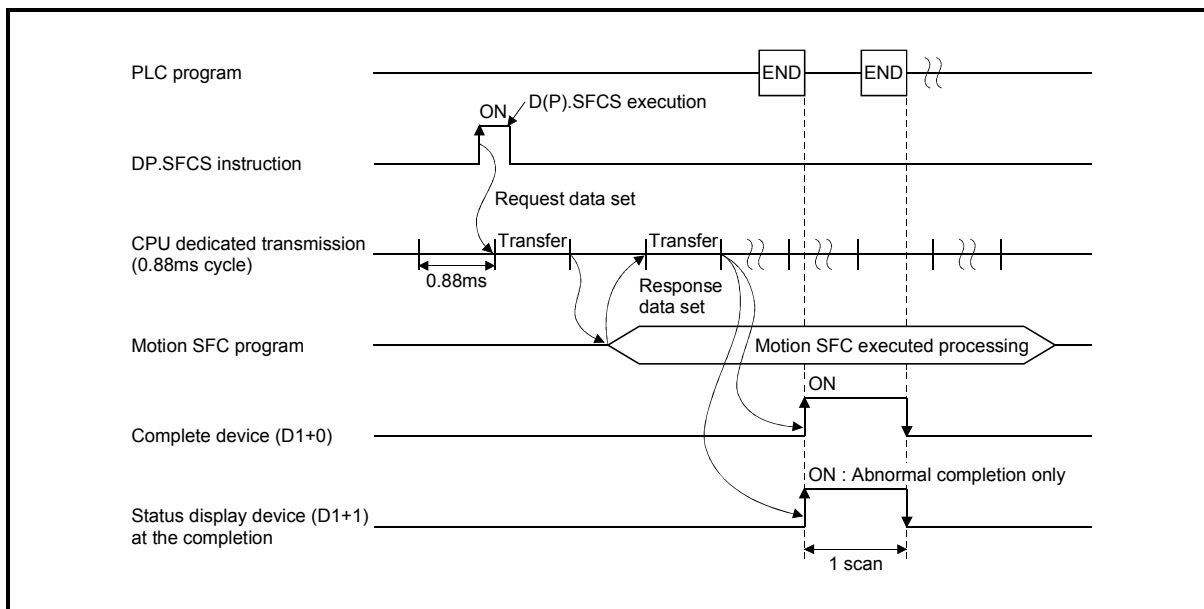
[Controls]

- (1) Request to start the Motion SFC program of program No. specified with (n2). The Motion SFC program can start any task setting of the normal task, event task and NMI task.
- (2) This instruction is always valid regardless of the state of real mode/virtual mode/ mode switching when the operating system software of Motion CPU is SV22.



[Operation]

Outline operation between CPUs at the DP.SFCS instruction execution is shown below.



3 MOTION DEDICATED PLC INSTRUCTION

[Setting range]

- (1) Setting of Motion SFC program

(n2) usable range
0 to 255

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).

If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status ^(Note) (Error code) (H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct a PLC program.
2100	There are 65 or more simultaneous D(P).SFCS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	
2200	The Motion SFC program No. to start is outside the range of 0 to 255.	

(Note) : 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

Error code ^(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct a PLC program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device which cannot be used for the instruction specified is specified.	

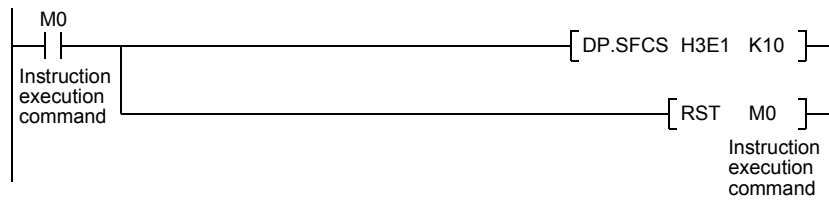
(Note) : 0 (Normal)

3 MOTION DEDICATED PLC INSTRUCTION

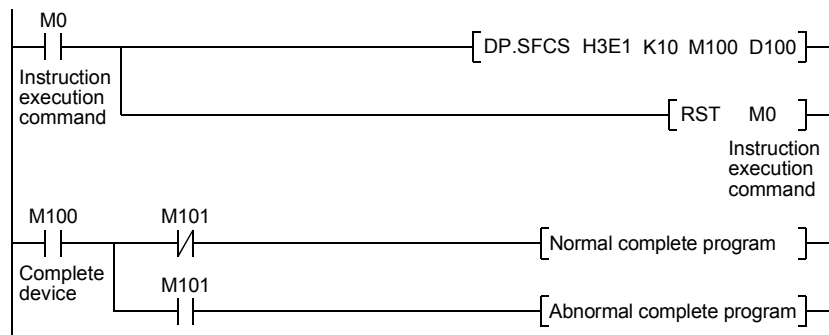
[Program example]

Program which starts the Motion SFC program of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.



3 MOTION DEDICATED PLC INSTRUCTION

3.2.2 Servo program start request from the PLC CPU to the Motion CPU:D(P).SVST (PLC instruction: D(P).SVST)

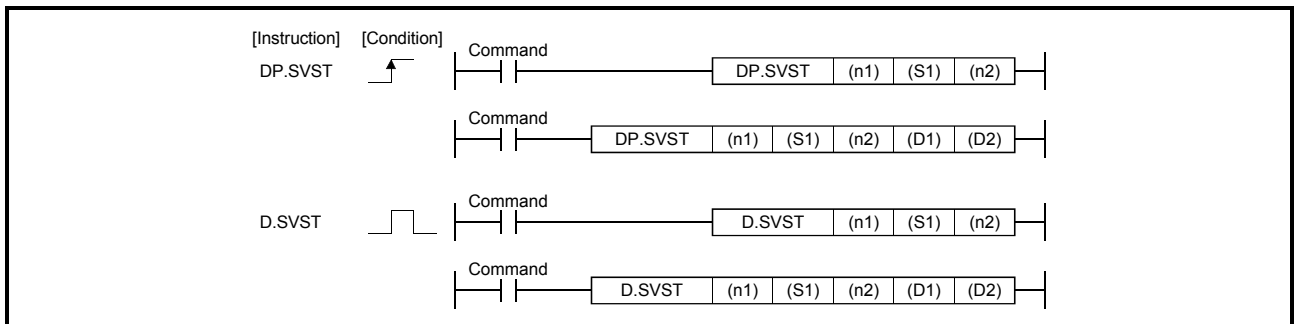
Setting data (Note-3)	Usable devices															
	Internal devices (System, User)				File register		Indirect specified	Link direct device J□\G			Unit access device U□\G□		Index register Z□	Constant		Others
	Bit		Word		Bit	Word		Bit	Word		Bit	Word		Decimal, Hexadecimal K, H	Real character string	
			Digit specified A	Digit specified B					Digit specified							
(n1)		○	○	○		○	○							○		
(S1)		○				○	○								○	
(n2)		○	○	○		○	○							○		
(D1) (Note-1)	△ (Note-2)				△ (Note-2)											
(D2) (Note-1)		△ (Note-2)				△ (Note-2)	○									

○ : Usable △ : Usable partly

(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : Local devices cannot be used.

(Note-3) : Setting data (n1) to (D2) : Index qualification possible



3 MOTION DEDICATED PLC INSTRUCTION

[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No.("Jn") ^(Note-2) to start. Q173DCPU : J1 to J32/Q172DCPU : J1 to J8	User	Character sequence
(n2)	Servo program No. to execute	User	16-bit binary
(D1) ^(Note-1)	Complete devices (D1+0) : Device which make turn on for one scan at accept completion of instruction. (D1+1) : Device which make turn on for one scan at accept abnormal completion of instruction. (("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) ^(Note-1)	Complete status storage device	System	Word

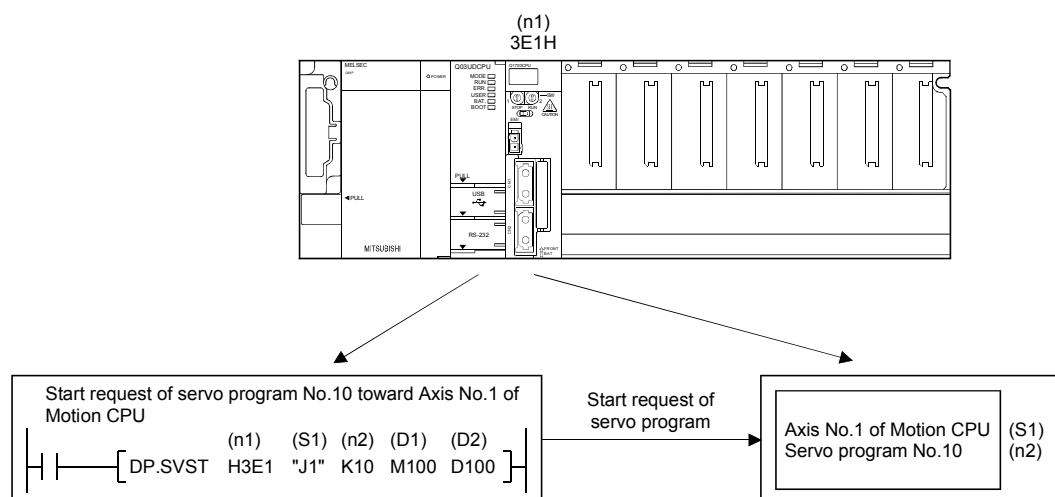
(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : "n" shows the numerical value correspond to axis No..

Q173DCPU : Axis No.1 to No.32 (n=1 to 32) / Q172DCPU : Axis No.1 to No.8 (n=1 to 8)

[Controls]

- (1) Request to start the servo program specified with (n2).
- (2) It is necessary to take an inter-lock by the start accept flag of CPU shared memory and user device so that multiple instructions may not be executed toward the same axis of the same Motion CPU No..



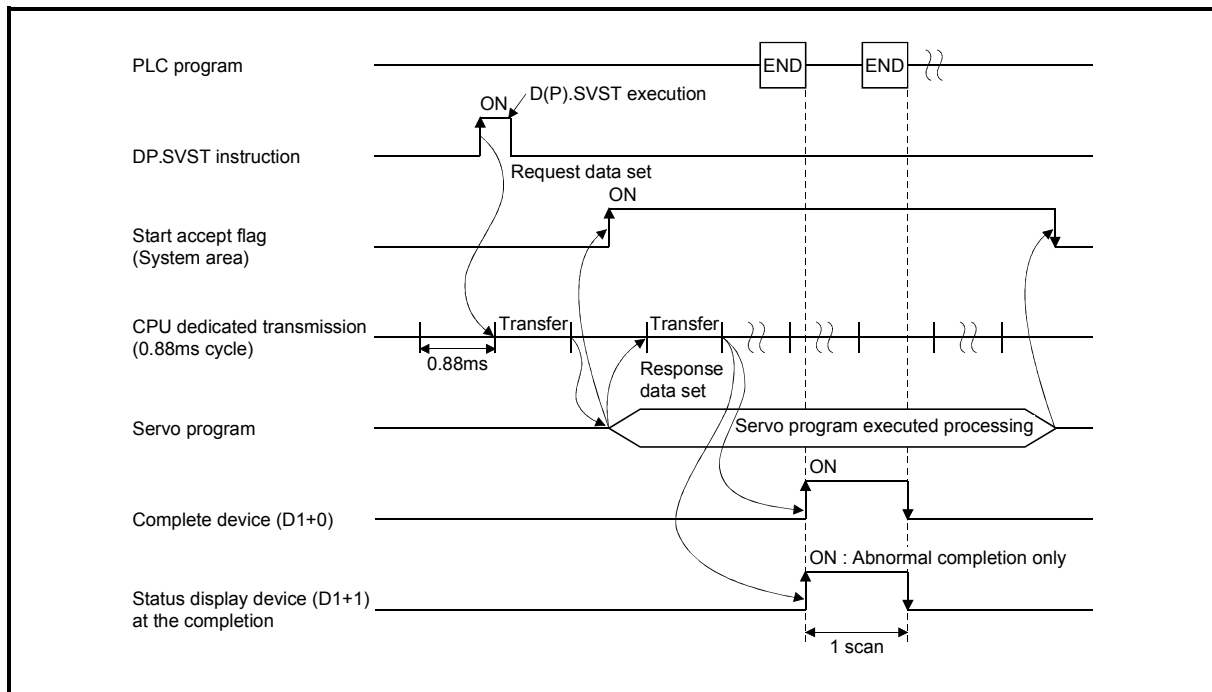
POINT

Refer to Section "3.3 Precautions" for details of the start accept flag.

3 MOTION DEDICATED PLC INSTRUCTION

[Operation]

Outline operation between CPUs at the DP.SVST instruction execution is shown below.



[Setting range]

(1) Setting of the starting axis

The starting axis set as (S1) is set J + Axis No. in a character sequence " ".

	(S1) usable range
Q173DCPU	1 to 32
Q172DCPU	1 to 8

Up to 8 axes can be set. Set them without dividing in a space etc. for multiple axes setting.

Set "J" in a capital letter and use the axis No. set in the system setting as the axis No. to start.

Refer to the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for system settings.

And, the axis No. to start does not need to be a order.

Example) When multiple axes (Axis1, Axis2, Axis10, Axis11) are set.
"J1J2J10J11"

(2) Setting of the servo program No.

(n2) usable range
0 to 4095

3 MOTION DEDICATED PLC INSTRUCTION

[Start accept flag (System area)]

The complete status of start accept flag is stored in the address of start accept flag in the CPU shared memory for target CPU.

CPU shared memory address () is decimal address	Description
204H(516) 205H(517)	The start accept flag for 32 axes are stored corresponding to each bit. (As for a bit's actually being set Q173DCPU : J1 to J32/Q172DCPU : J1 to J8.) OFF : Start accept enable ON : Start accept disable

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).

If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status ^(Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct PLC program.
2100	There are 65 or more simultaneous D(P).SVST/D(P).CHGA sum table instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	
2201	The servo program No. to execute is outside the range of 0 to 4095.	
2202	Axis No. set by D(P).SVST instruction is wrong.	

(Note) : 0000H (Normal)

3 MOTION DEDICATED PLC INSTRUCTION

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

Error code ^(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct PLC program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note) : 0 (Normal)

3 MOTION DEDICATED PLC INSTRUCTION

3.2.3 Current value change instruction from the PLC CPU to the Motion CPU: D(P).CHGA (PLC instruction: D(P).CHGA)

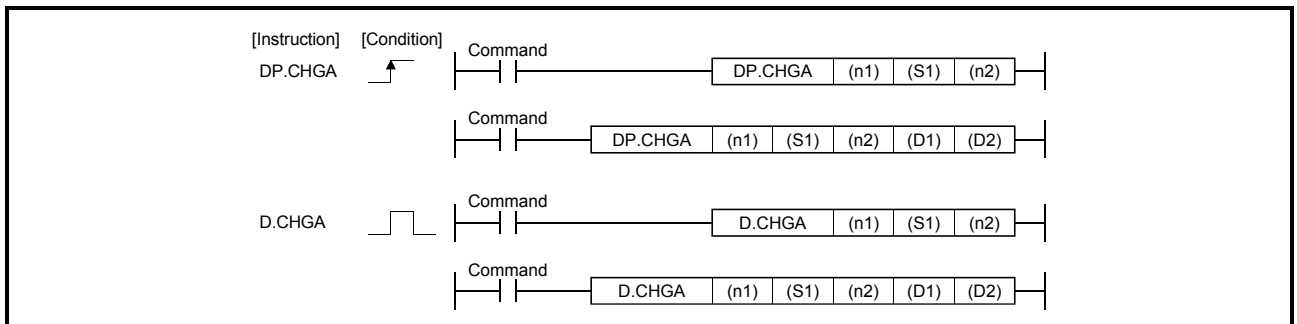
Setting data (Note-3)	Usable devices															
	Internal devices (System, User)				File register		Indirect specified	Link direct device J□\G			Unit access device U□\G□		Index register Z□	Constant		Others
	Bit		Word		Bit	Word		Bit	Word		Bit	Word		Decimal, Hexadecimal K, H	Real character string	
			Digit specified A	Digit specified B					Digit specified							
(n1)		○	○	○		○	○							○		
(S1)		○				○	○								○	
(n2)		○	○	○		○	○							○		
(D1) (Note-1)	△ (Note-2)				△ (Note-2)											
(D2) (Note-1)		△ (Note-2)				△ (Note-2)	○									

○ : Usable △ : Usable partly

(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : Local devices cannot be used.

(Note-3) : Setting data (n1) to (D2) : Index qualification possible



3 MOTION DEDICATED PLC INSTRUCTION

[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No. ("Jn") ^(Note-2) to execute the current value change. Q173DCPU : J1 to J32/Q172DCPU : J1 to J8	User	Character sequence
	Synchronous encoder axis No. ("En") ^(Note-3) to execute the current value change. Q173DCPU : E1 to E12/Q172DCPU : E1 to E8		
	Cam axis No. ("Cn") ^(Note-2) to execute the current value change within 1 revolution. Q173DCPU : C1 to C32/Q172DCPU : C1 to C8		
(n2)	Current value to change	User	32-bit binary
(D1) ^(Note-1)	Complete devices (D1+0) : Device which make turn on for one scan at accept completion of instruction. (D1+1) : Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) ^(Note-1)	Complete status storage device	System	Word

(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : "n" shows the numerical value correspond to axis No..

Q173DCPU : Axis No.1 to No.32 (n=1 to 32) / Q172DCPU : Axis No.1 to No.8 (n=1 to 8)

(Note-3) : "n" shows the numerical value correspond to synchronous encoder axis No..

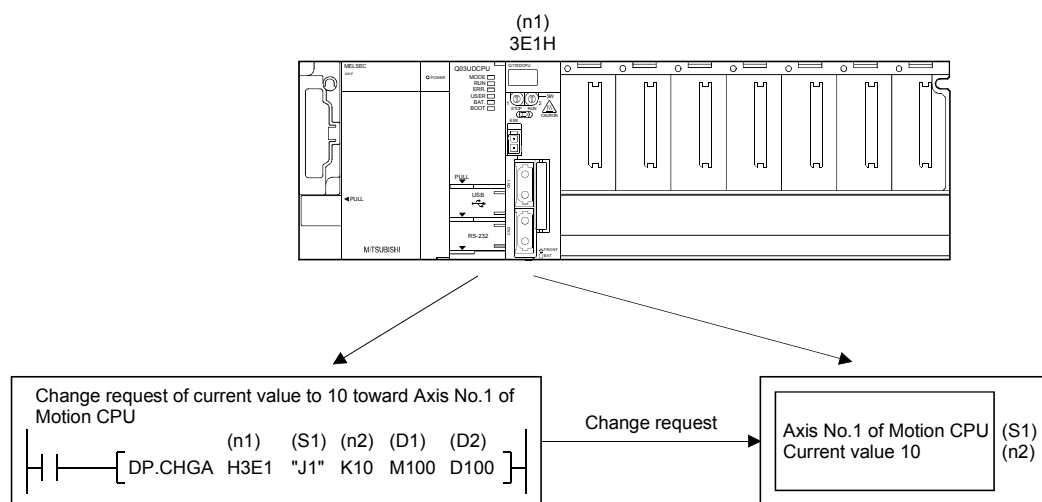
Q173DCPU : Axis No.1 to No.12 (n=1 to 12) / Q172DCPU : Axis No.1 to No.8 (n=1 to 8)

3 MOTION DEDICATED PLC INSTRUCTION

● When axis No."Jn" is specified with (S1)

[Controls]

- (1) The current value change of axis (stopped axis) specified with (S1) is changed to the current value specified with (n2).
- (2) It is necessary to take an inter-lock by the start accept flag and user device of CPU shared memory so that multiple instructions may not be executed toward the same axis of same Motion CPU.
- (3) The current change value is also possible when the servo program which makes the CHGA instruction toward an axis is executed in the D(P).SVST instruction.



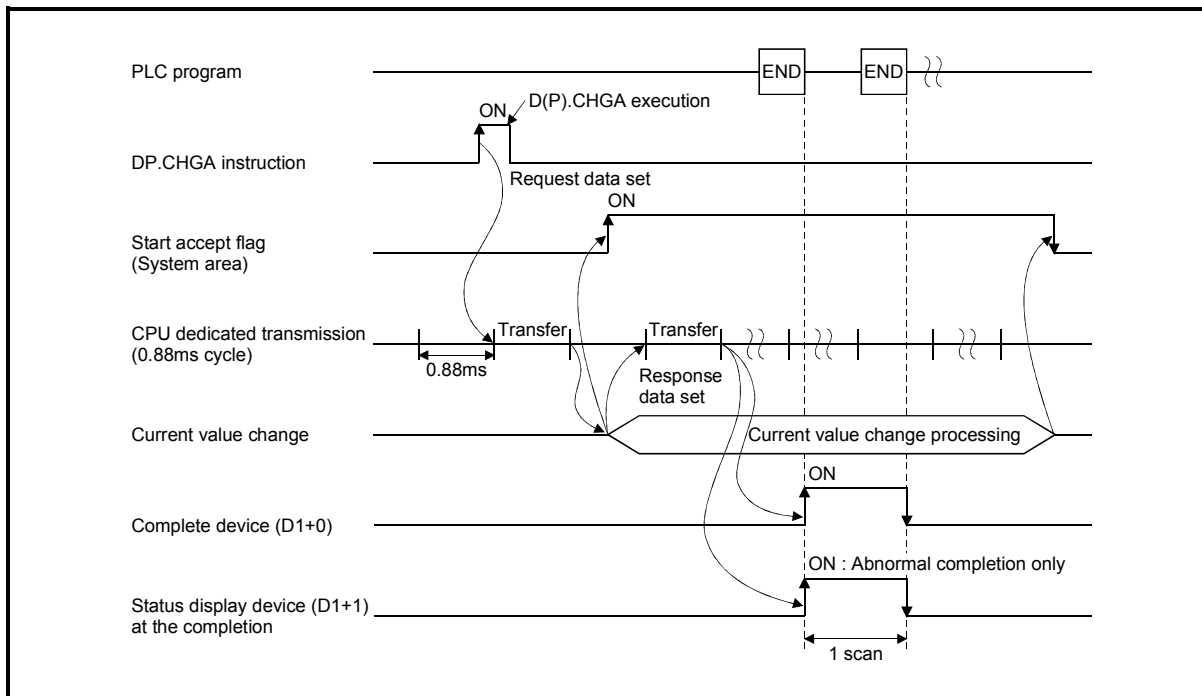
POINT

Refer to Section "3.3 Precautions" for details of the start accept flag.

3 MOTION DEDICATED PLC INSTRUCTION

[Operation]

Outline operation between CPUs at the DP.CHGA instruction execution by specifying "Jn" as Axis No. is shown below.



[Setting range]

(1) Setting of axis to execute the current value change

The axis to execute the current value change set as (S1) sets J + Axis No. in a character sequence " ".

	(S1) usable range
Q173DCPU	1 to 32
Q172DCPU	1 to 8

The number of axes which can set are only 1 axis.

Set "J" in a capital letter and use the axis No. set in the system setting as the axis No. to start.

Refer to the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for system settings.

(2) Setting of the current value to change

(n2) usable range
-2147483648 to 2147483647

3 MOTION DEDICATED PLC INSTRUCTION

[Start accept flag (System area)]

When the instruction is executed by specifying "Jn" as Axis No., the complete status of start accept flag is stored in the address of the start accept flag in the CPU shared memory for target CPU.

CPU shared memory address () is decimal address	Description
204H(516) 205H(517)	The start accept flag for 32 axes are stored corresponding to each bit. (As for a bit's actually being set Q173DCPU : J1 to J32^Q172DCPU : J1 to J8.) OFF : Start accept enable ON : Start accept disable

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).

If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status ^(Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct PLC program.
2100	There are 65 or more simultaneous D(P).SVST/D(P).CHGA sum table instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	
2203	Axis No. set by D(P).CHGA instruction is wrong.	

(Note) : 0000H (Normal)

3 MOTION DEDICATED PLC INSTRUCTION

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

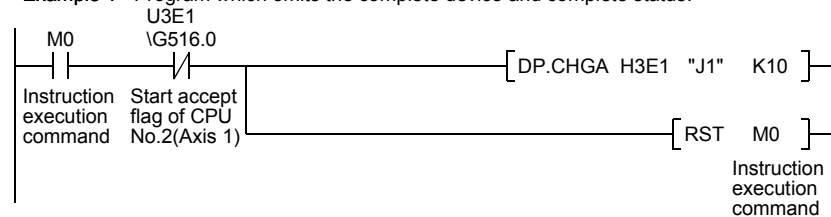
Error code ^(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct PLC program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note) : 0 (Normal)

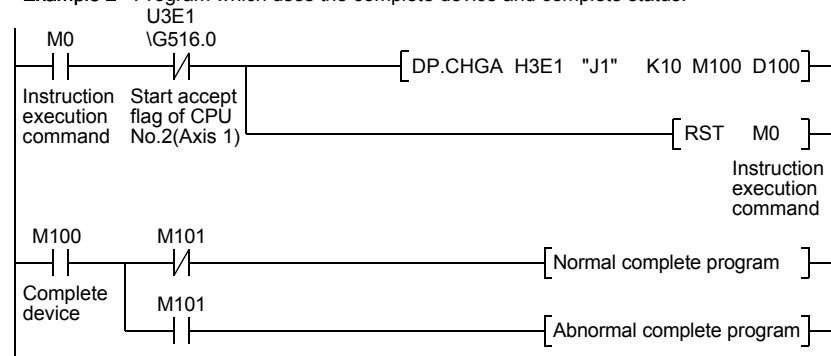
[Program example]

Program which changes the current value to 10 for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.

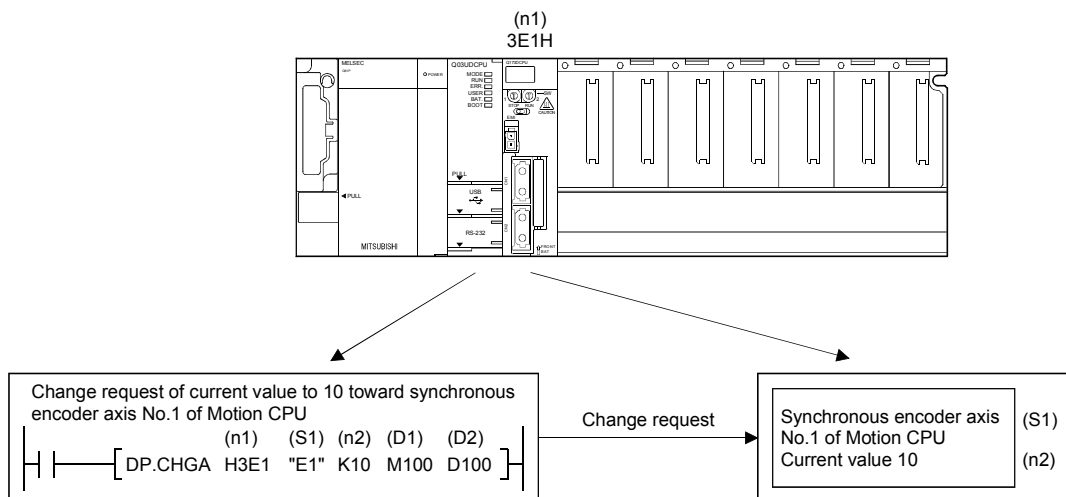


3 MOTION DEDICATED PLC INSTRUCTION

● When axis No."En" is specified with (S1)

[Controls]

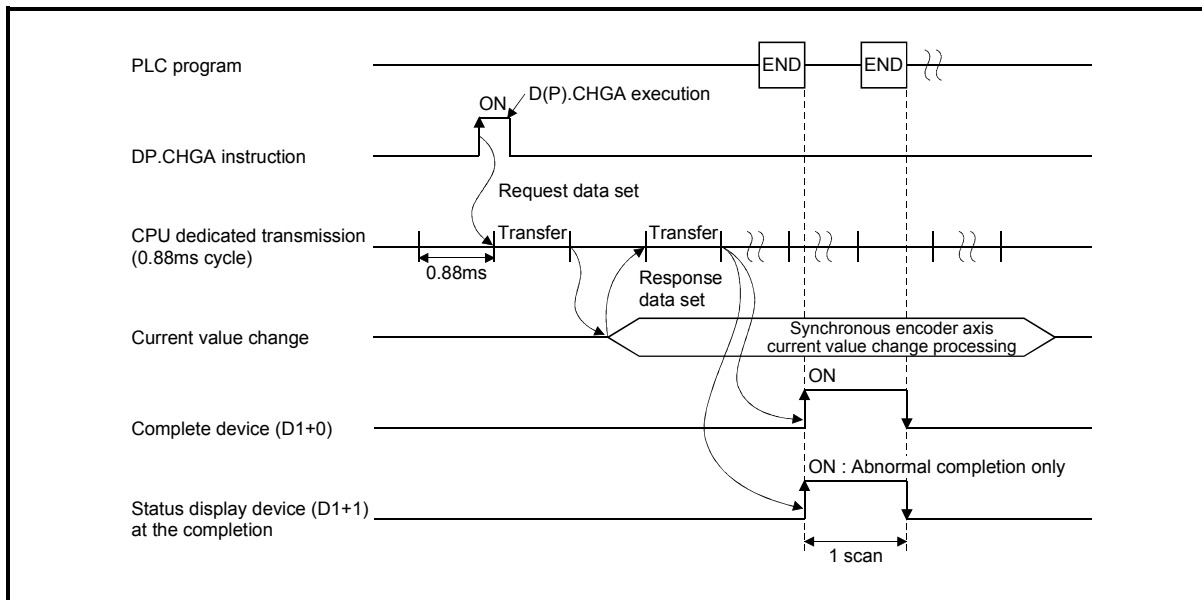
- (1) The synchronous encoder axis current value specified with (S1) is changed to the current value specified with (n2) in the virtual mode.
- (2) There is not an interlock signal for status of synchronous encoder current value change.
When the multiple instructions are executed toward the same synchronous encoder axis of same Motion CPU, the current value is changed to specified value by last instruction.
- (3) The current change value is also possible when the servo program which makes the CHGA instruction toward the synchronous encoder axis is executed in the S(P).SVST instruction.



3 MOTION DEDICATED PLC INSTRUCTION

[Operation]

Outline operation between CPUs at the DP.CHGA instruction execution by specifying "En" as Axis No. is shown below.



[Setting range]

- (1) Setting of synchronous encoder axis to execute the current value change
The synchronous encoder axis to execute the current value change set as (S1) sets E + synchronous encoder axis No. in a character sequence " ".

	(S1) usable range
Q173DCPU	1 to 12
Q172DCPU	1 to 8

The number of axes which can set are only 1 axis.

Set "E" in a capital letter and use the axis No. set in the system setting as the axis No. to start.

Refer to the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for system settings.

- (2) Setting of the current value to change

(n2) usable range
-2147483648 to 2147483647

3 MOTION DEDICATED PLC INSTRUCTION

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).

If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status ^(Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct PLC program.
2002	A Motion dedicated PLC instruction that does not correspond with the operating system of the Motion CPU was executed. (Axis No. "En" was specified by operating system software except SV22.)	
2100	There are 65 or more simultaneous D(P).SVST/D(P).CHGA sum table instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	
2203	Axis No. set by D(P).CHGA instruction is wrong.	

(Note) : 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

Error code ^(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct PLC program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

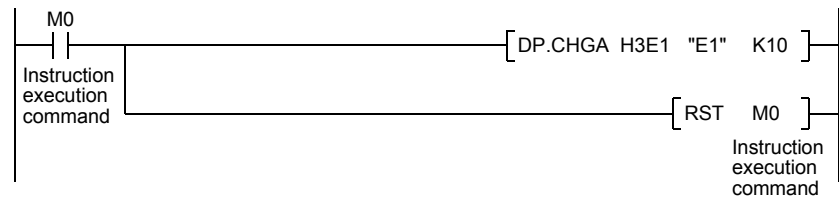
(Note) : 0 (Normal)

3 MOTION DEDICATED PLC INSTRUCTION

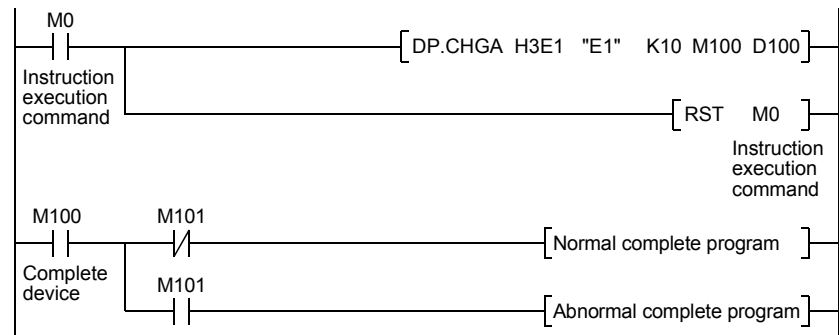
[Program example]

Program which changes the current value to 10 for synchronous encoder axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.

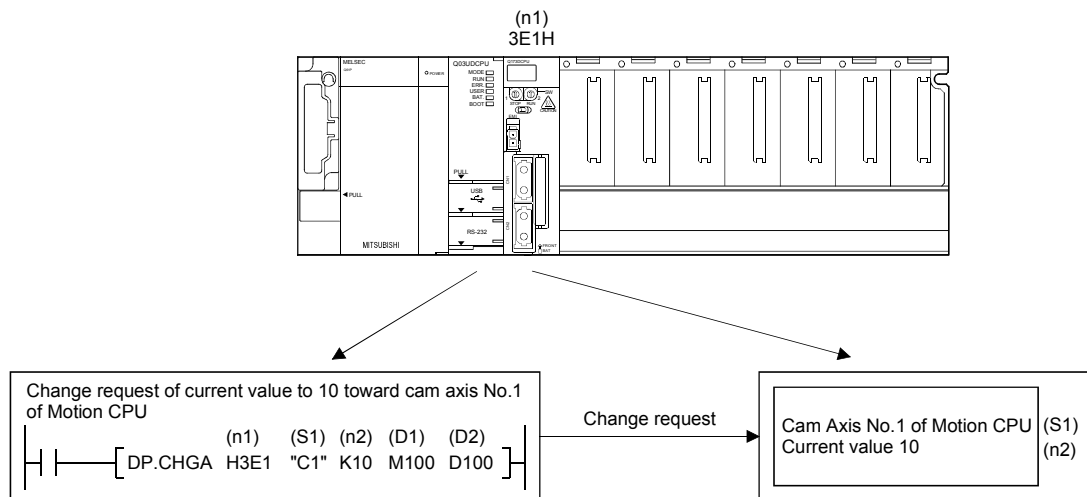


3 MOTION DEDICATED PLC INSTRUCTION

● When axis No. "Cn" is specified with (S1)

[Controls]

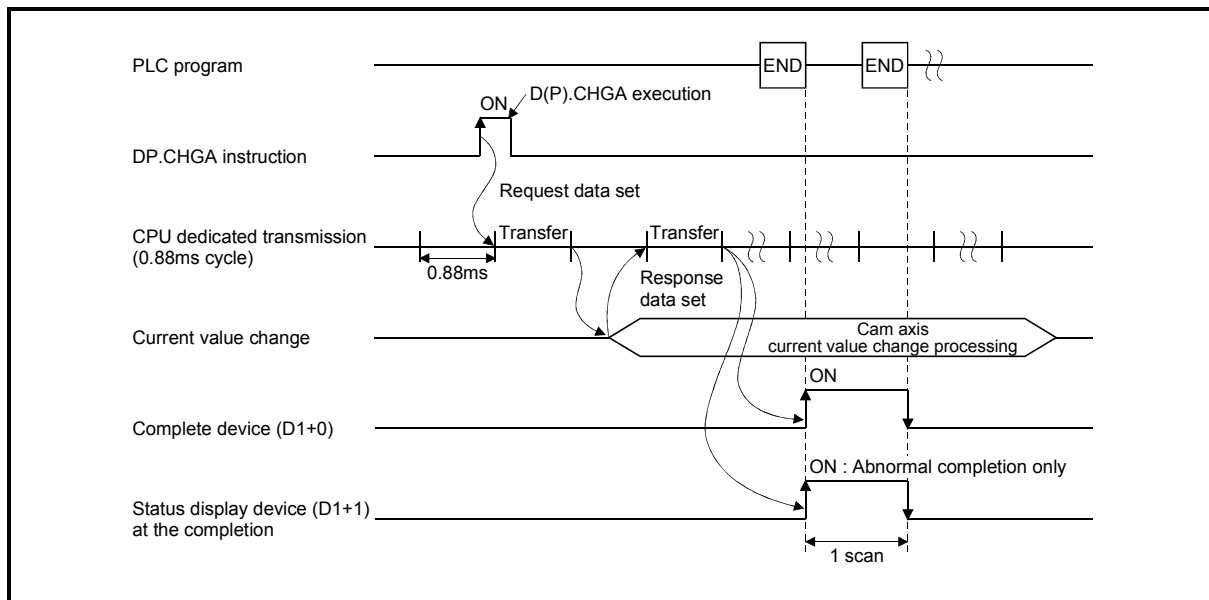
- (1) The current value within 1 cam shaft revolution specified with (S1) is changed to the current value specified with (n2) in the virtual mode.
- (2) There is not an interlock signal for status of current value within 1 cam shaft revolution change.
When the multiple instructions are executed toward the same cam axis of same Motion CPU, the current value is changed to specified value by last instruction.
- (3) The current change value is also possible when the servo program which makes the CHGA instruction toward the cam axis is executed in the S(P).SVST instruction.



3 MOTION DEDICATED PLC INSTRUCTION

[Operation]

Outline operation between CPUs at the DP.CHGA instruction execution by specifying "Cn" as Axis No. is shown below.



[Setting range]

- (1) Setting of cam axis to execute the current value change within 1 cam shaft revolution

The cam axis to execute the current value change within 1 cam shaft revolution set as (S1) sets C + cam axis No. in a character sequence " ".

	(S1) usable range
Q173DCPU	1 to 32
Q172DCPU	1 to 8

The number of axes which can set are only 1 axis.

Set "C" in a capital letter and use the axis No. set in the system setting as the axis No. to start.

Refer to the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for system settings.

- (2) Setting of the current value to change

(n2) usable range
-2147483648 to 2147483647

3 MOTION DEDICATED PLC INSTRUCTION

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).

If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status ^(Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct PLC program.
2002	A Motion dedicated PLC instruction that does not correspond with the operating system of the Motion CPU was executed. (Axis No. "En" was specified by operating system software except SV22.)	
2100	There are 65 or more simultaneous D(P).SVST/D(P).CHGA sum table instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	
2203	Axis No. set by D(P).CHGA instruction is wrong.	

(Note) : 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

Error code ^(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct PLC program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

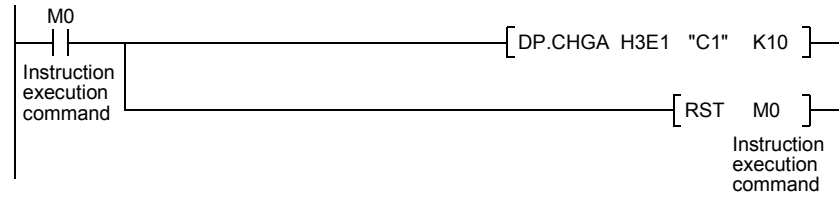
(Note) : 0 (Normal)

3 MOTION DEDICATED PLC INSTRUCTION

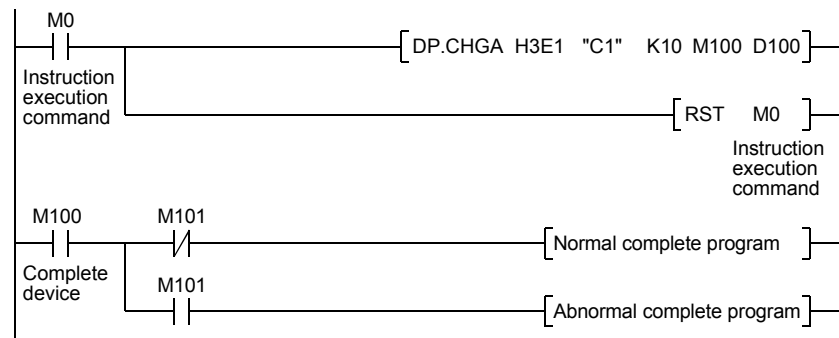
[Program example]

Program which changes the current value to 10 for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.



3 MOTION DEDICATED PLC INSTRUCTION

3.2.4 Speed change instruction from the PLC CPU to the Motion CPU: D(P).CHGV

(PLC instruction: D(P).CHGV)

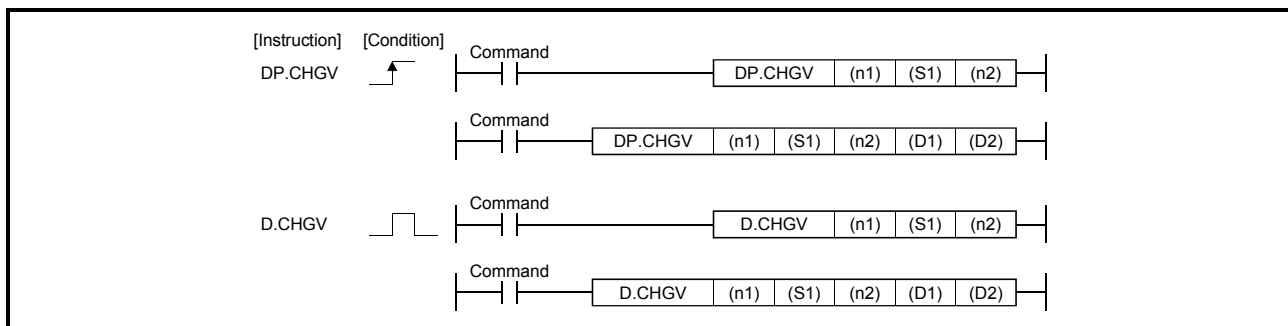
Setting data (Note-3)	Usable devices															
	Internal devices (System, User)				File register		Indirect specified	Link direct device J□\G			Unit access device U□\G□		Index register Z□	Constant		Others
	Bit		Word		Bit	Word		Bit	Word		Bit	Word		Decimal, Hexadecimal K, H	Real character string	
			Digit specified A	Digit specified B					Digit specified							
(n1)		○	○	○		○	○							○		
(S1)		○				○	○								○	
(n2)		○	○	○		○	○							○		
(D1) (Note-1)	△ (Note-2)				△ (Note-2)											
(D2) (Note-1)		△ (Note-2)				△ (Note-2)	○									

○ : Usable △ : Usable partly

(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : Local devices cannot be used.

(Note-3) : Setting data (n1) to (D2) : Index qualification possible



3 MOTION DEDICATED PLC INSTRUCTION

[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No. ("Jn") ^(Note-2) to execute the speed change. Q173DCPU : J1 to J32/Q172DCPU : J1 to J8	User	Character sequence
(n2)	Speed to change	User	32-bit binary
(D1) ^(Note-1)	Complete devices (D1+0) : Device which make turn on for one scan at accept completion of instruction. (D1+1) : Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) ^(Note-1)	Complete status storage device	System	Word

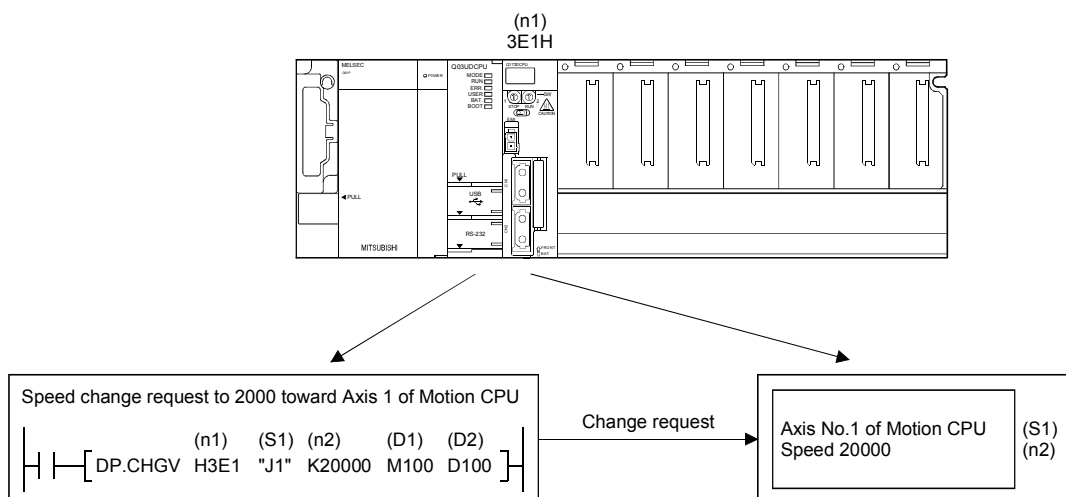
(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : "n" shows the numerical value correspond to axis No..

Q173DCPU : Axis No.1 to No.32 (n=1 to 32) / Q172DCPU : Axis No.1 to No.8 (n=1 to 8)

[Controls]

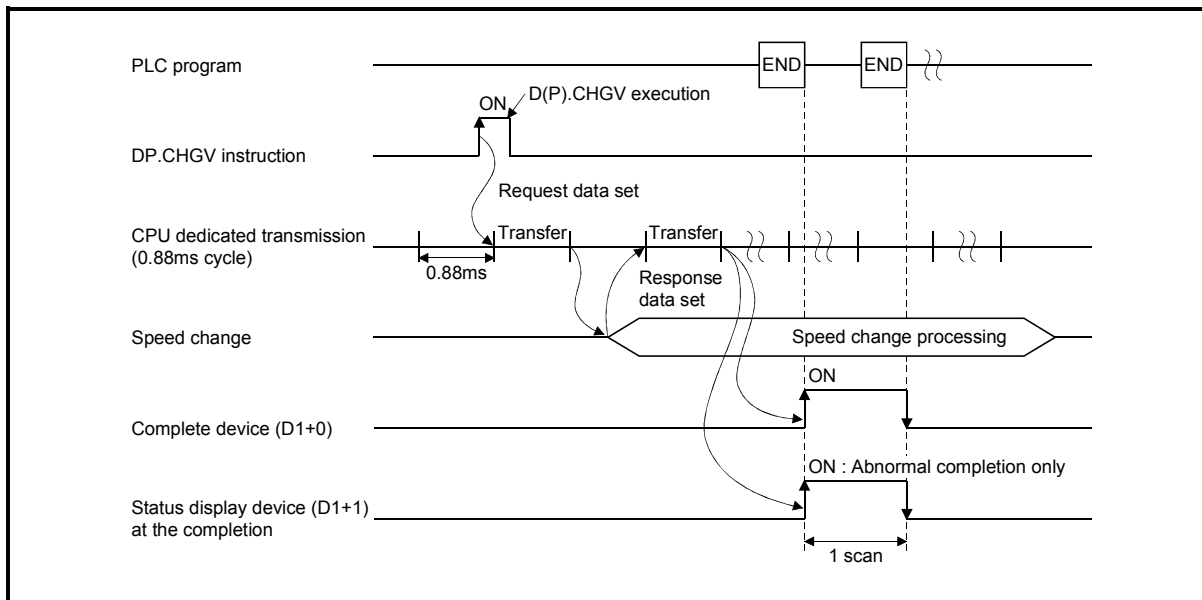
- (1) The speed of axis specified with (S1) is changed to the speed specified with (n2) during positioning or JOG operating.
- (2) There is not an interlock signal on the shared memory during speed change.
When the multiple instructions are executed toward the same axis of same Motion CPU, the speed is changed to specified value by last instruction.



3 MOTION DEDICATED PLC INSTRUCTION

[Operation]

Outline operation between CPUs at the DP.CHGV instruction execution is shown below.



[Setting range]

(1) Setting of axis to execute the speed change

The axis to execute the speed change set as (S1) sets J + axis No. in a character sequence " ".

	(S1) usable range
Q173DCPU	1 to 32
Q172DCPU	1 to 8

The number of axes which can set are only 1 axis.

Set "J" in a capital letter and use the axis No. set in the system setting as the axis No. to start.

Refer to the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for system settings.

(2) Setting of the speed to change

(n2) usable range
-2147483648 to 2147483647

3 MOTION DEDICATED PLC INSTRUCTION

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).

If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status ^(Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct PLC program.
2204	Axis No. set by D(P).CHGV instruction is wrong.	

(Note) : 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

Error code ^(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct PLC program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

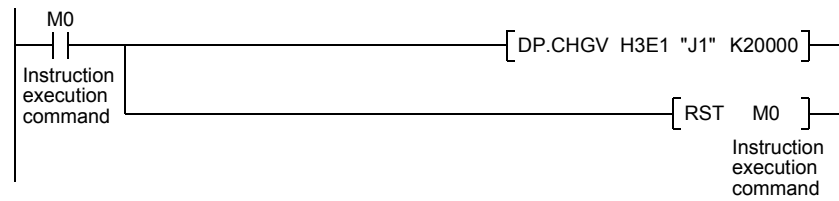
(Note) : 0 (Normal)

3 MOTION DEDICATED PLC INSTRUCTION

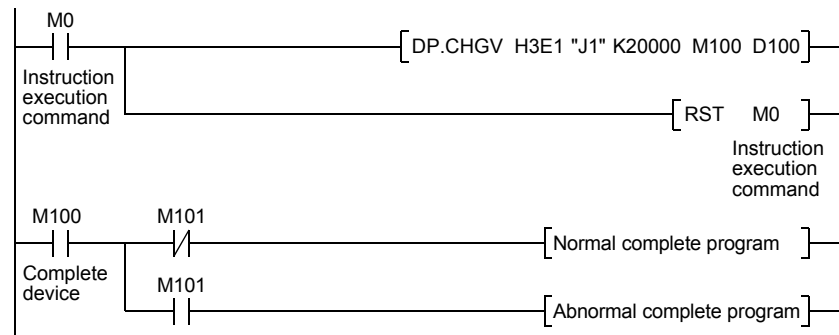
[Program example]

Program which changes the positioning speed to 20000 for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.

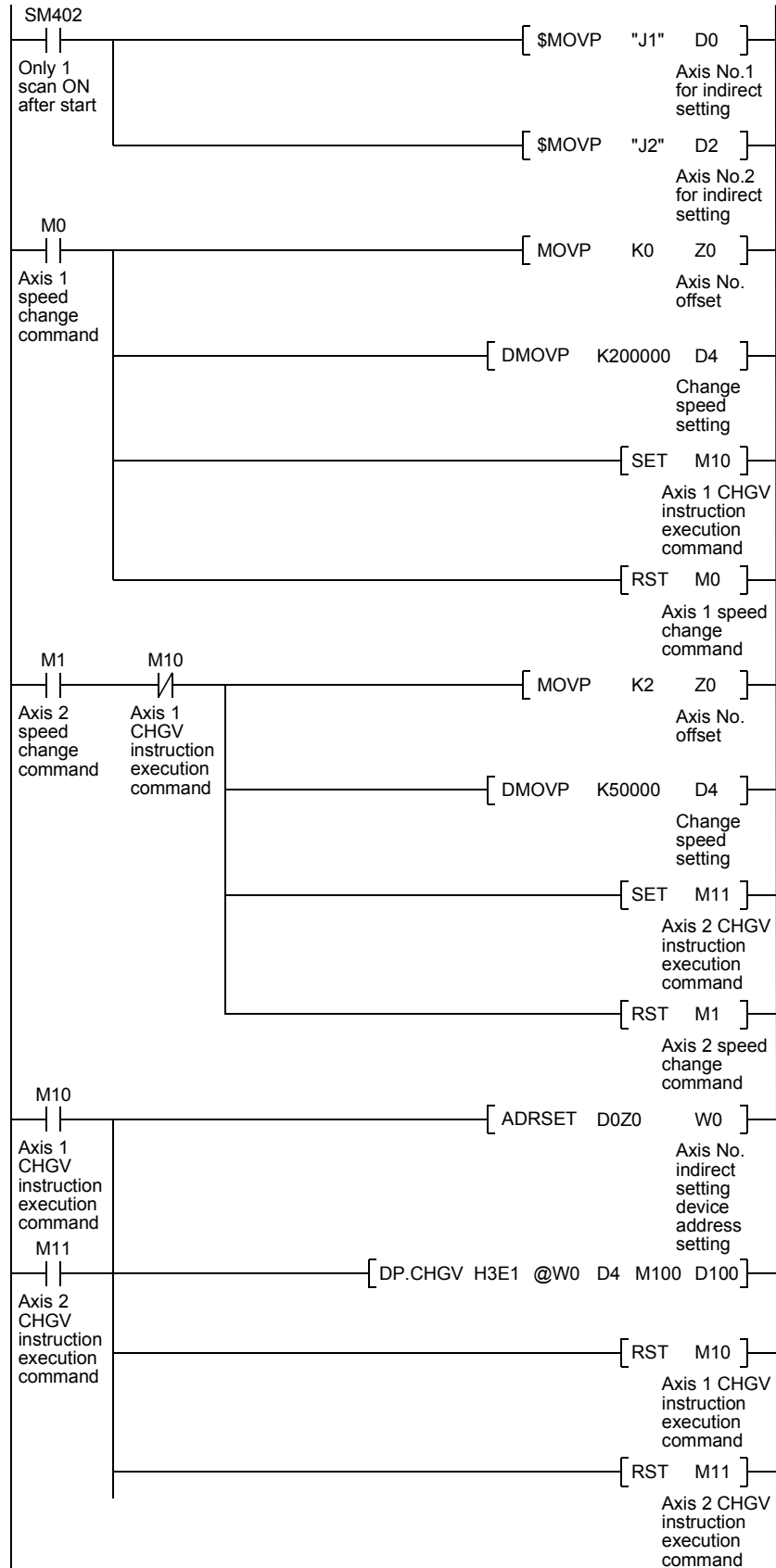


<Example 2> Program which uses the complete device and complete status.



3 MOTION DEDICATED PLC INSTRUCTION

Program which changes the positioning speed to 200000 for Axis 1 of the Motion CPU (CPU No.2), when M0 that sets Axis No. as indirect setting method turned ON, and then changes the positioning speed to 50000 for Axis 2, when M1 turned ON.



3 MOTION DEDICATED PLC INSTRUCTION

3.2.5 Torque limit value change request instruction from the PLC CPU to the Motion CPU:

D(P).CHGT (PLC instruction: D(P).CHGT)

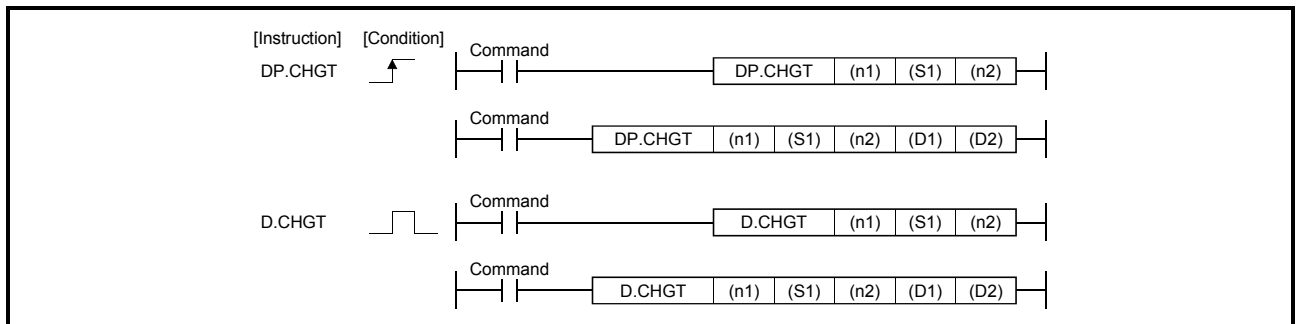
Setting data (Note-3)	Usable devices															
	Internal devices (System, User)				File register		Indirect specified	Link direct device J□\G			Unit access device U□\G□		Index register Z□	Constant		Others
	Bit		Word		Bit	Word		Bit	Word		Bit	Word		Decimal, Hexadecimal K, H	Real character string	
			Digit specified A	Digit specified B					Digit specified							
(n1)		○	○	○		○	○							○		
(S1)		○				○	○								○	
(n2)		○	○	○		○	○							○		
(D1) (Note-1)	△ (Note-2)				△ (Note-2)											
(D2) (Note-1)		△ (Note-2)				△ (Note-2)	○									

○ : Usable △ : Usable partly

(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : Local devices cannot be used.

(Note-3) : Setting data (n1) to (D2) : Index qualification possible



3 MOTION DEDICATED PLC INSTRUCTION

[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No. ("Jn") ^(Note-2) to execute the torque limit value change. Q173DCPU : J1 to J32/Q172DCPU : J1 to J8	User	Character sequence
(n2)	Torque limit value to change	User	16-bit binary
(D1) ^(Note-1)	Complete devices (D1+0) : Device which make turn on for one scan at accept completion of instruction. (D1+1) : Device which make turn on for one scan at accept abnormal completion of instruction. (("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) ^(Note-1)	Complete status storage device	System	Word

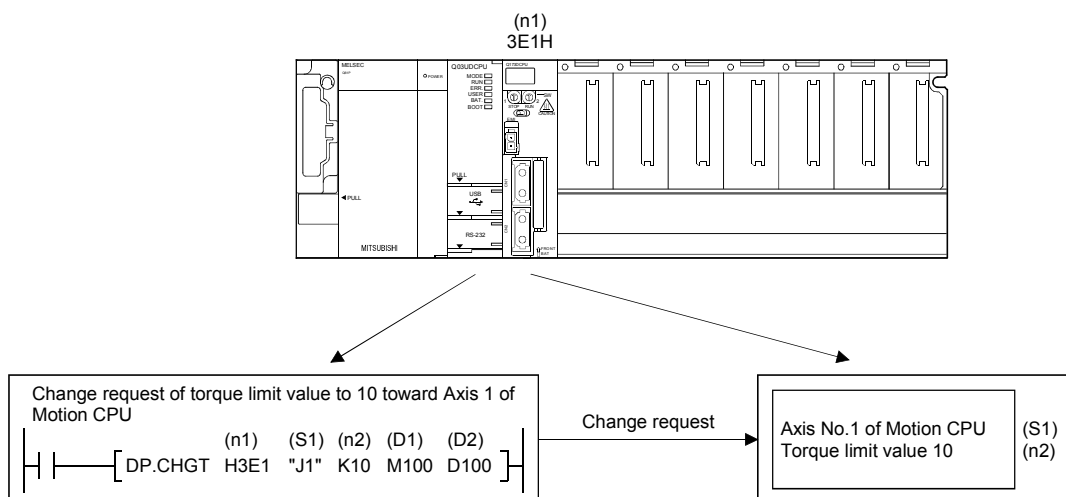
(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : "n" shows the numerical value correspond to axis No..

Q173DCPU : Axis No.1 to No.32 (n=1 to 32) / Q172DCPU : Axis No.1 to No.8 (n=1 to 8)

[Controls]

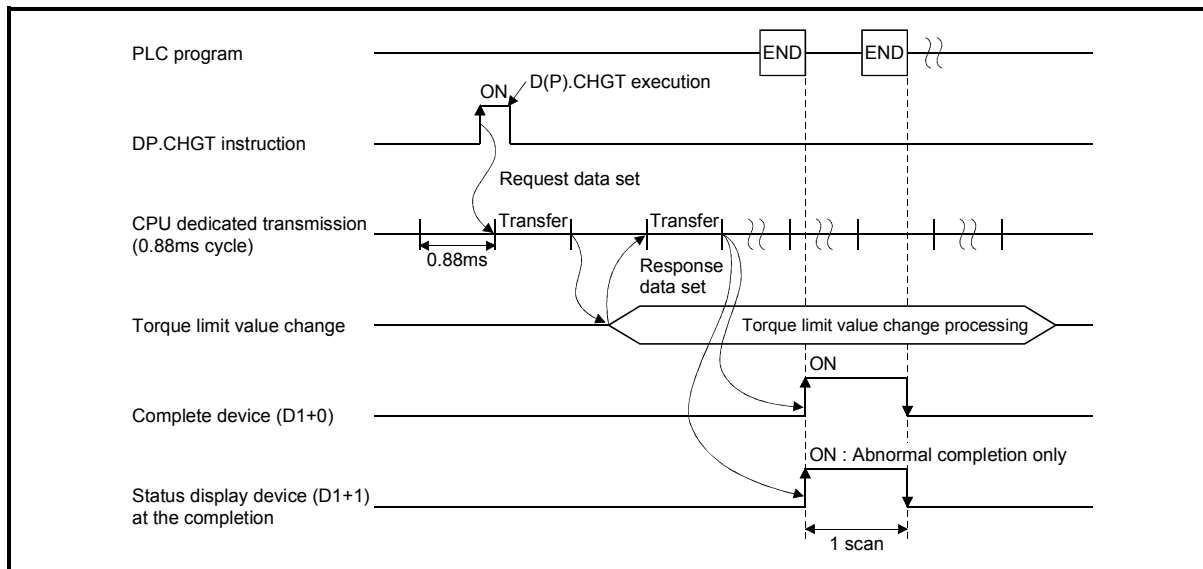
- (1) The torque limit value of axis specified with (S1) is changed to the value specified with (n2) regardless of while being operating or stopping 1n the real mode.
- (2) There is not an interlock signal for status of axis torque change.
When the multiple instructions are executed toward the same axis of same Motion CPU, the torque is changed to specified value by last instruction.



3 MOTION DEDICATED PLC INSTRUCTION

[Operation]

Outline operation between CPUs at the DP.CHGT instruction execution is shown below.



[Setting range]

(1) Setting of axis to execute the torque limit value change

The axis to execute the torque limit change set as (S1) sets J + axis No. in a character sequence " ".

	(S1) usable range
Q173DCPU	1 to 32
Q172DCPU	1 to 8

The number of axes which can set are only 1 axis.

Set "J" in a capital letter and use the axis No. set in the system setting as the axis No. to start.

Refer to the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for system settings.

(2) Setting of the torque limit value to change

(n2) usable range
1 to 1000

3 MOTION DEDICATED PLC INSTRUCTION

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).

If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status ^(Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct PLC program.
2205	Axis No. set by D(P).CHGT instruction is wrong.	

(Note) : 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

Error code ^(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct PLC program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

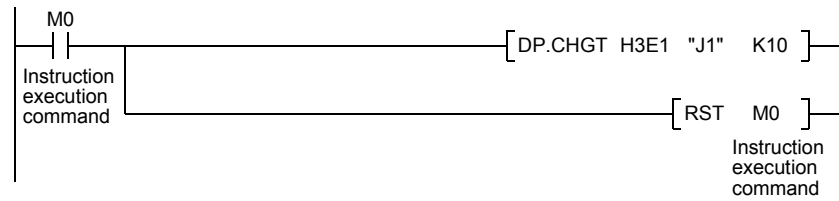
(Note) : 0 (Normal)

3 MOTION DEDICATED PLC INSTRUCTION

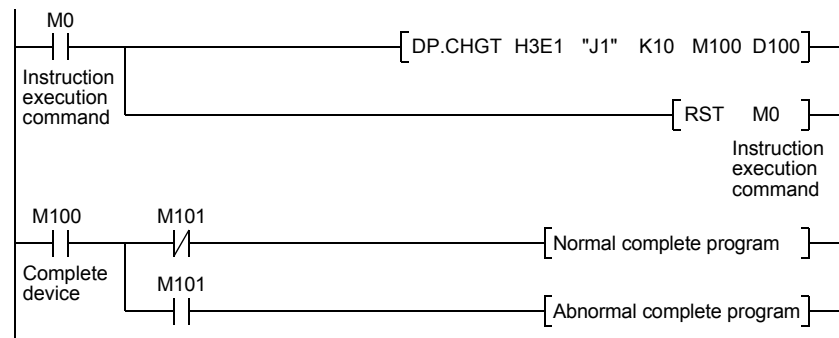
[Program example]

Program which changes the torque limit value to 10[%] for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.



3 MOTION DEDICATED PLC INSTRUCTION

3.2.6 Write device data of the self CPU to the device of other CPU: D(P).DDWR

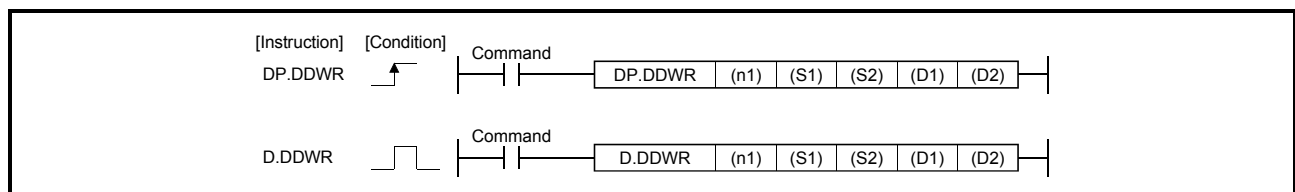
(PLC instruction: D(P).DDWR)

Setting data (Note-2)	Usable devices															
	Internal devices (System, User)				File register		Indirect specified	Link direct device J□\G			Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word			Bit	Word		Bit	Word		Bit	Word		Decimal, Hexadecimal K, H	Real character string	
		Digit specified A	Digit specified B	Digit specified												
(n1)		○	○	○		○	○							○		
(S1)		△ (Note-1)				△ (Note-1)	○									
(S2)		○	○			○	○									
(D1)		○	○			○	○								○	
(D2)	△ (Note-1)				△ (Note-1)											

○ : Usable △ : Usable partly

(Note-1) : Local devices cannot be used.

(Note-2) : Setting data (n1) to (D2) : Index qualification possible



[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Start device of the self CPU where control data are stored.	User	Word
(S2)	Start device of the self CPU where writing data are stored.	User	Word
(D1)	Start device of the target Motion CPU that stores writing data. POINT Data can be written in device like a motion register (#) etc. of Motion CPU outside the range in the PLC CPU that executes this instruction, by setting it by a character sequence " ".	User	Word/ Character sequence
(D2)	Complete devices (D2+0) : Device which make turn on for one scan at accept completion of instruction. (D2+1) : Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit

3 MOTION DEDICATED PLC INSTRUCTION

[Control data]

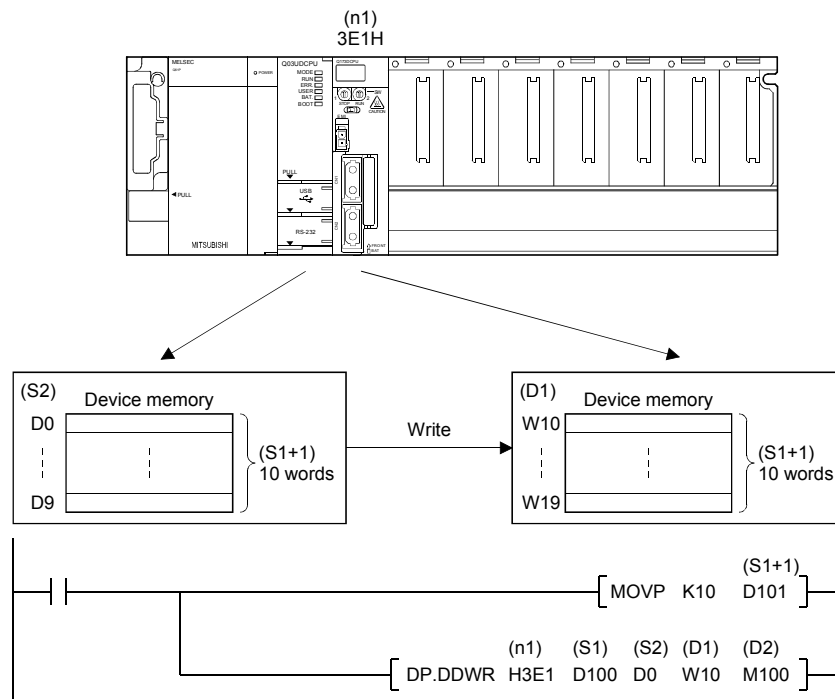
Device	Item	Setting data	Setting range	Set by
S1+0	Complete status	The status at the instruction completion is stored. 0 : No error (Normal completion) Except 0 : Error code	—	System
S1+1	Number of writing data	Set the number of writing data with each word	1 to 20	User

[Controls]

- (1) A part for the number of writing data of the control data specified with (S1+1) of data since the device specified with (S2) of the self CPU are stored to since the word device specified with (D1) of the target CPU (n1) in the Multiple CPU system.
- (2) Figure specification of the bit device is possible for (S2) and (D1). However, figure specification is 4 figures and a start bit device number is only the multiple of 16. It becomes MULTI-COM.ERROR (Error code: 4353) when other values are specified.
- (3) Data can be written in device like a motion register (#) etc. of Motion CPU outside the range in the PLC CPU that executes this instruction, by setting (D1) by a character sequence " ".
- (4) D(P).DDWR instruction accepting and normal/abnormal completion can be confirmed with the complete device (D2) or status display device (D2+1) at the completion.
 - (a) Complete device
It is turned on by the END processing of scan which the instruction completed, and turned off by the next END processing.
 - (b) Status display device at the completion
It is turned on/off according to the status of the instruction completion.
 - Normal completion : OFF
 - Abnormal completion : It is turned on by the END processing of scan which the instruction completed, and turned off by the next END processing.
(Error code is stored in control data (S1+0: Complete status).)

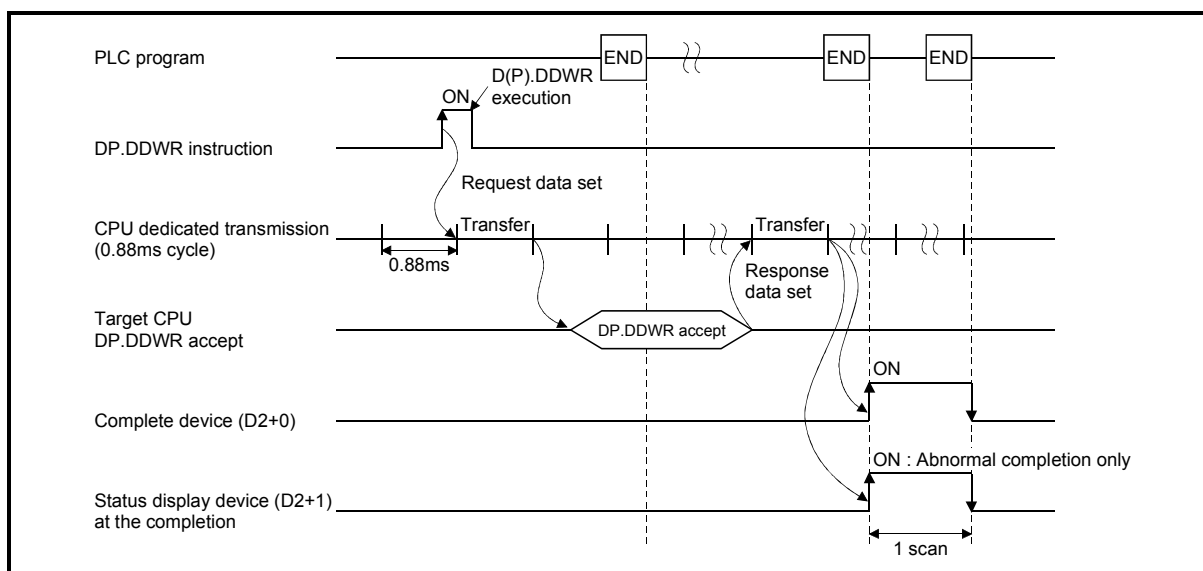
3 MOTION DEDICATED PLC INSTRUCTION

- (5) There is a limitation for number of simultaneous instruction execution/
simultaneous acceptance in the Motion dedicated PLC instruction.
Exchange a large amount of data through the CPU shared memory.



[Operation]

Outline operation between CPUs at the DP.DDWR instruction execution is shown below.



3 MOTION DEDICATED PLC INSTRUCTION

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (S0+0).

Complete status ^(Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct PLC program.
2001	The specified device cannot be used in the Motion CPU, or it is outside the device range.	
2080	Number of writing data points set by D(P).DDWR instruction is wrong.	
2100	There are 65 or more simultaneous D(P).DDRD/D(P).DDWR sum table instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	

(Note) : 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

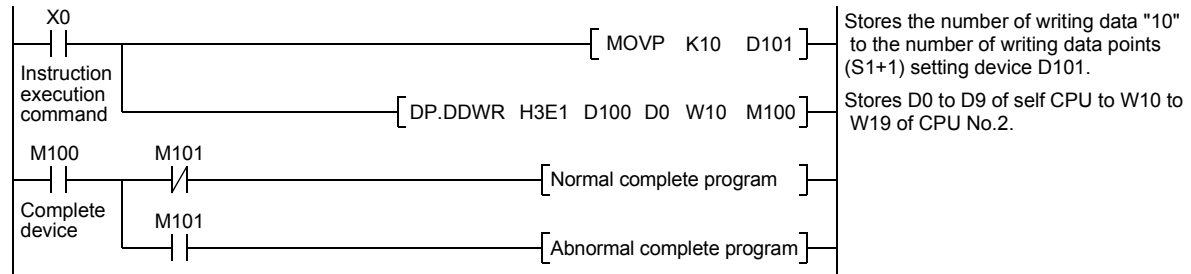
Error code ^(Note)	Error factor	Corrective action
4101	Number of writing data exceeded the range of storage device.	Confirm a program, and correct it to a correct PLC program.
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	
	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
	The number of devices for instruction specified is wrong.	
4351	The device that cannot be used by the instruction specified is specified.	
4352	The character string that cannot be handled by the instruction specified is specified.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	
4355	Number of writing data is outside the range of 1 to 20.	

(Note) : 0 (Normal)

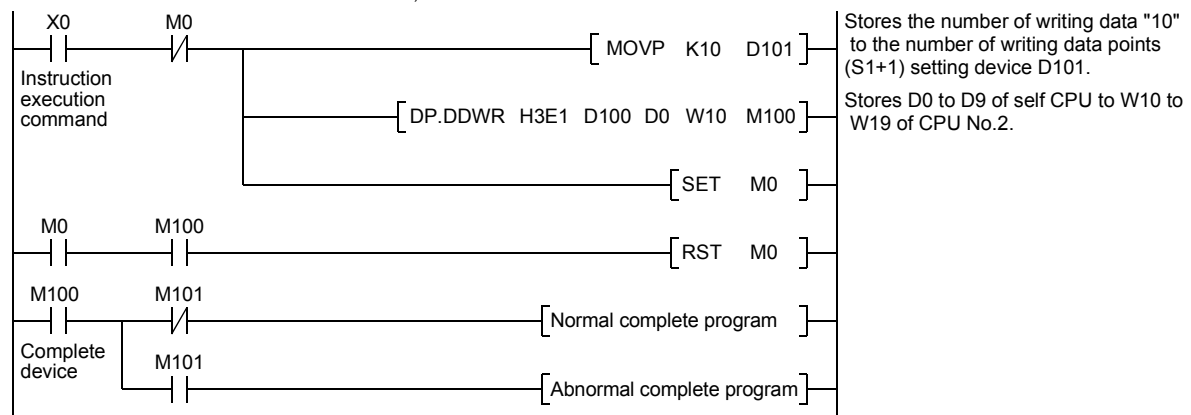
3 MOTION DEDICATED PLC INSTRUCTION

[Program example]

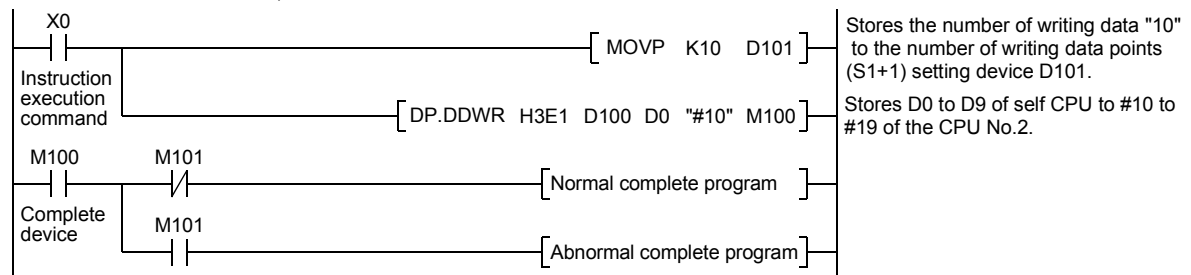
<Example 1> Program which stores data for 10 words from D0 of the self CPU to W10 or later of the CPU No.2, when X0 turned ON.



<Example 2> Program which stores simultaneously data for 10 words from D0 of the self CPU to W10 or later of the CPU No.2, while X0 is ON.



<Example 3> Program which stores data for 10 words from D0 of the self CPU to #10 or later of the CPU No.2, when X0 turned ON.



3 MOTION DEDICATED PLC INSTRUCTION

3.2.7 Read device data of other CPU to the device of self CPU:D(P).DDRD

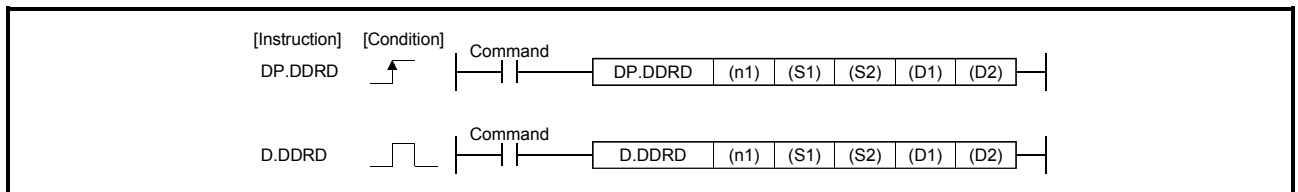
(PLC instruction: D(P).DDRD)

Setting data (Note-2)	Usable devices															
	Internal devices (System, User)				File register		Indirect specified	Link direct device J□\G			Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word			Bit	Word		Bit	Word		Bit	Word		Decimal, Hexadecimal K, H	Real character string	
		Digit specified A	Digit specified B	Digit specified												
(n1)		○	○	○		○	○							○		
(S1)		△ (Note-1)				△ (Note-1)	○									
(S2)		○	○			○	○								○	
(D1)		△ (Note-1)	△ (Note-1)			△ (Note-1)	○									
(D2)	△ (Note-1)				△ (Note-1)											

○ : Usable △ : Usable partly

(Note-1) : Local devices cannot be used.

(Note-2) : Setting data (n1) to (D2) : Index qualification possible



[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Start device of the self CPU where control data is stored.	User	Word
(S2)	Start device of the target CPU where reading data is stored. POINT Data can be read from device like a motion register (#) etc. of Motion CPU outside the range in the PLC CPU that executes this instruction, by setting it by a character sequence " ".	User	Word/ Character sequence
(D1)	Start device of the self CPU which stores the reading data.	User	Word
(D2)	Complete devices (D2+0) : Device which make turn on for one scan at accept completion of instruction. (D2+1) : Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit

3 MOTION DEDICATED PLC INSTRUCTION

[Control data]

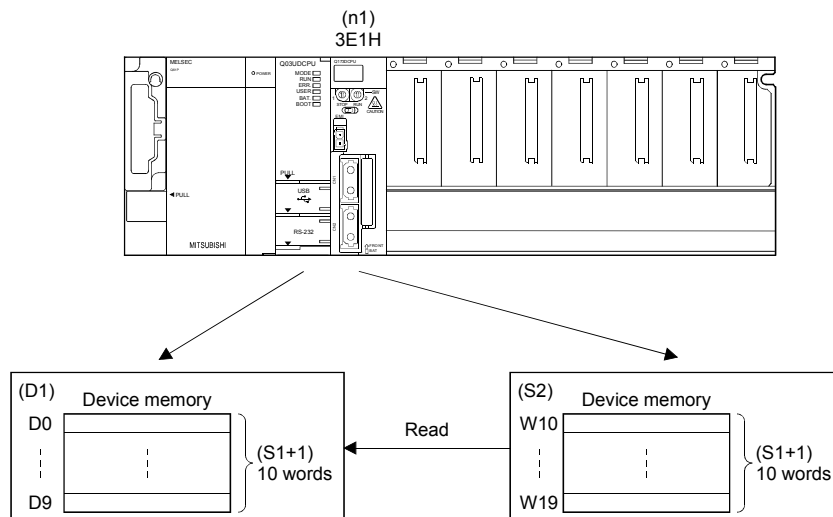
Device	Description	Setting data	Setting range	Set by
S1+0	Complete status	The status at the instruction completion is stored. 0 : No error (Normal completion) Except 0 : Error code	—	System
S1+1	Number of reading data	Set the number of reading data.	1 to 20	User

[Controls]

- (1) A part for the number of reading data of the control data specified with (S1+1) of data since the device specified with (S2) in the target CPU (n1) is stored to since the word device specified with (D1) of the self CPU in the Multiple CPU system.
- (2) Figure specification of the bit device is possible for (S2) and (D1). However, figure specification is 4 figures and a start bit device number is only the multiple of 16. It becomes MULTI-COM.ERROR (Error code: 4353) when other values are specified.
- (3) Data can be read from device of the Motion CPU out side the range in the PLC CPU that executes the this instruction like a motion register (#) etc., by setting (S2) by a character sequence " ".
- (4) D(P).DDRD instruction accepting and normal/abnormal completion can be confirmed with the complete device (D2) or status display device (D2+1) at the completion.
 - (a) Complete device
It is turned on by the END processing of scan which the instruction completed, and turned off by the next END processing.
 - (b) Status display device at the completion
It is turned on/off according to the status of the instruction completion.
 - Normal completion : OFF
 - Abnormal completion : It is turned on by the END processing of scan which the instruction completed, and turned off by the next END processing.
(Error code is stored in control data (S1+0: Complete status).)

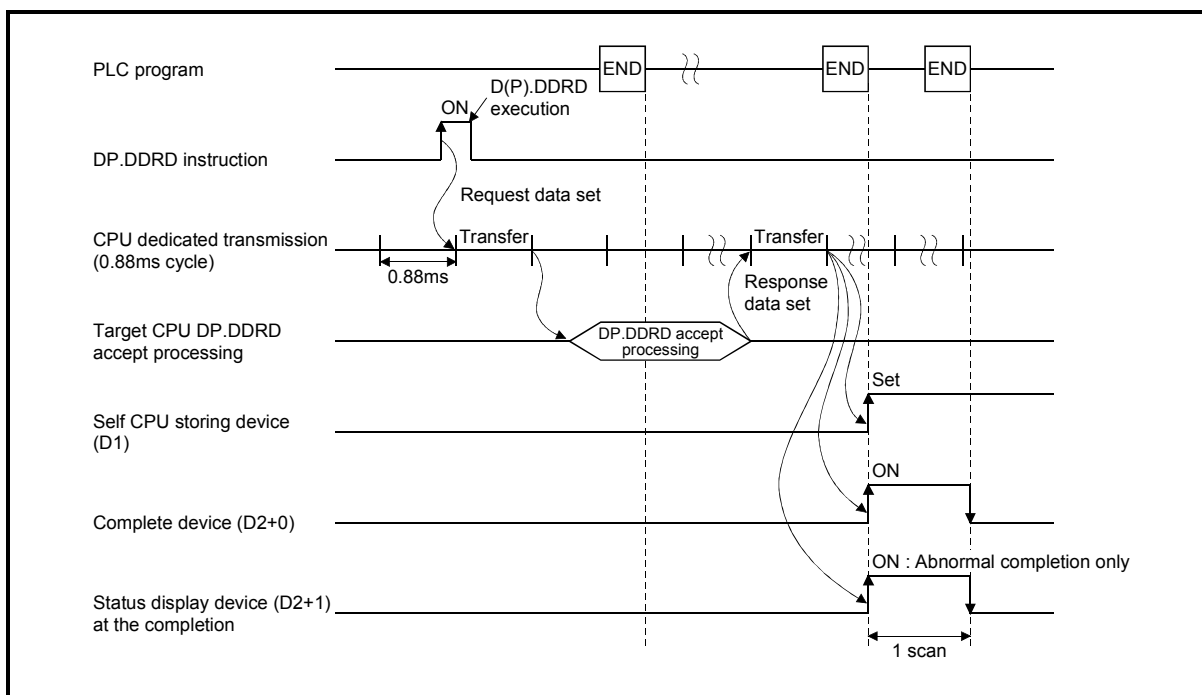
3 MOTION DEDICATED PLC INSTRUCTION

- (5) There is a limitation for number of simultaneous instruction execution/ simultaneous acceptance in the Motion dedicated PLC instruction.
Exchange a large amount of data through the CPU shared memory.



[Operation]

Outline operation between CPUs at the DP.DDRD instruction execution is shown below.



3 MOTION DEDICATED PLC INSTRUCTION

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (S0+0).

Complete status ^(Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct PLC program.
2001	The specified device cannot be used in the Motion CPU, or it is outside the device range.	
2081	Number of reading data points set by D(P).DDRD instruction is wrong.	
2100	There are 65 or more simultaneous D(P).DDRD/D(P).DDWR sum table instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	

(Note) : 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

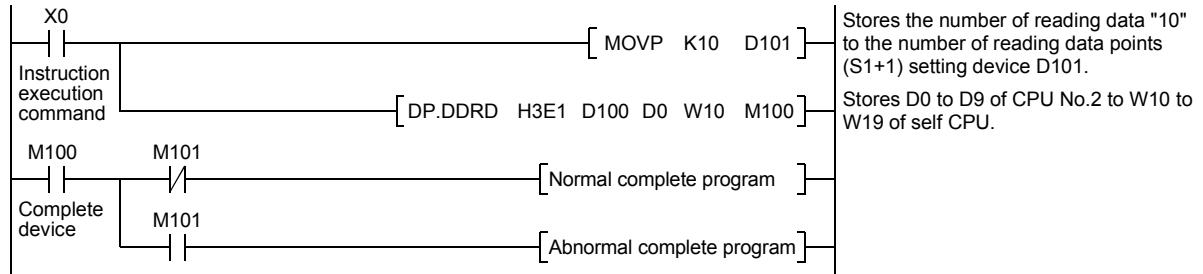
Error code ^(Note)	Error factor	Corrective action
4101	Number of writing data exceeded the range of storage device.	Confirm a program, and correct it to a correct PLC program.
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	
	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
	The number of devices for instruction specified is wrong.	
4351	The device that cannot be used by the instruction specified is specified.	
4352	The character string that cannot be handled by the instruction specified is specified.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	
4355	Number of writing data is outside the range of 1 to20.	

(Note) : 0 (Normal)

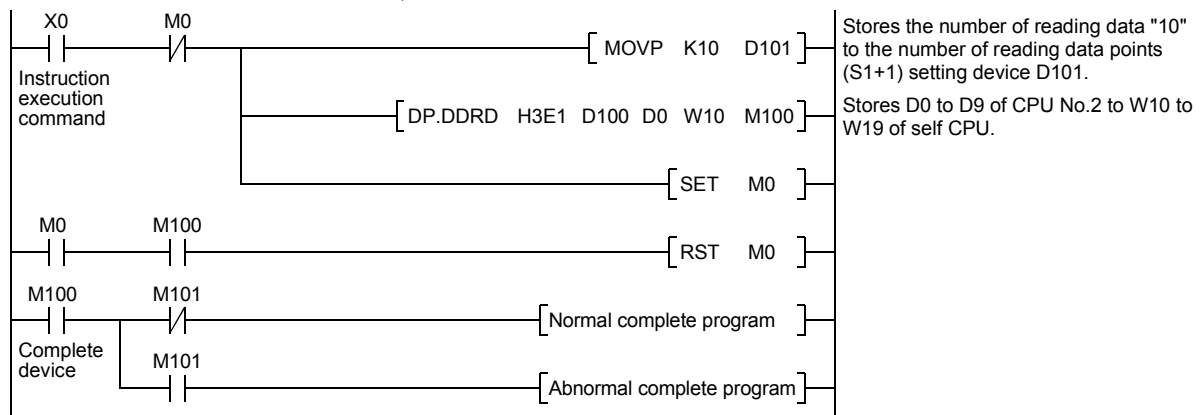
3 MOTION DEDICATED PLC INSTRUCTION

[Program example]

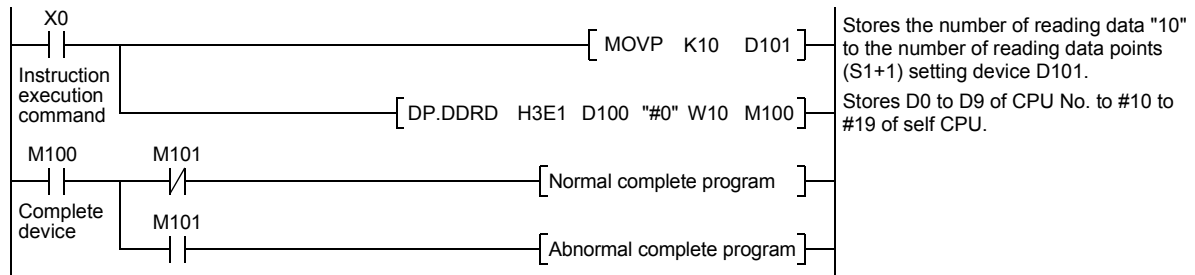
<Example 1> Program which stores data for 10 words from D0 of the CPU No.2 to W10 or later of the self CPU, when X0 turned ON.



<Example 2> Program which stores simultaneously data for 10 words from D0 of the CPU No.2 to W10 or later of the self CPU, while X0 turned ON.



<Example 3> Program which stores data for 10 words from D0 of the CPU No.2 to #10 or later of the self CPU, when X0 turned ON.



3 MOTION DEDICATED PLC INSTRUCTION

3.2.8 Interrupt instruction to the other CPU: D(P).GINT

(PLC instruction: D(P).GINT)

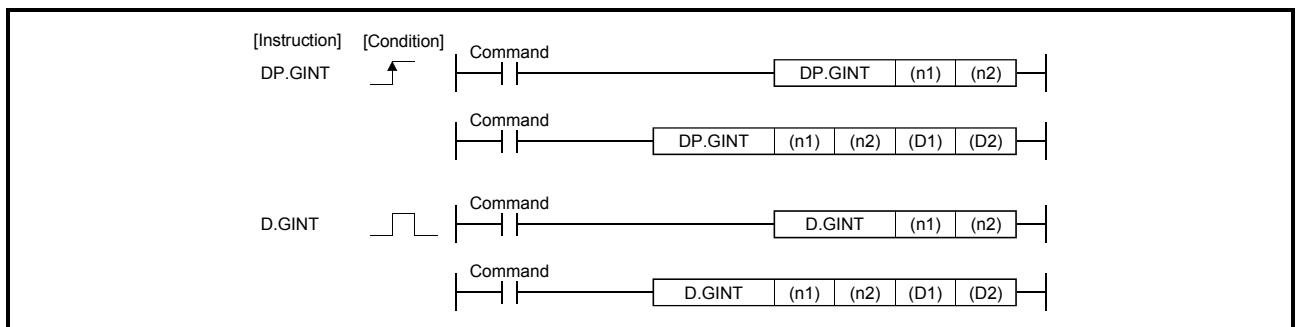
Setting data (Note-3)	Usable devices															
	Internal devices (System, User)				File register		Indirect specified	Link direct device J□\G			Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word		Bit	Word	Bit		Word		Bit	Word	Decimal, Hexadecimal K, H		Real character string		
		Digit specified A	Digit specified B					Digit specified								
(n1)		○	○	○		○	○							○		
(n2)		○	○	○		○	○							○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)		△ (Note-2)											
(D2) (Note-1)		△ (Note-2)				△ (Note-2)	○									

○ : Usable △ : Usable partly

(Note-1) : Omission possible with both of (D1) and (D2) omission.

(Note-2) : Local devices cannot be used.

(Note-3) : Setting data (n1) to (D2) : Index qualification possible



[Setting data]

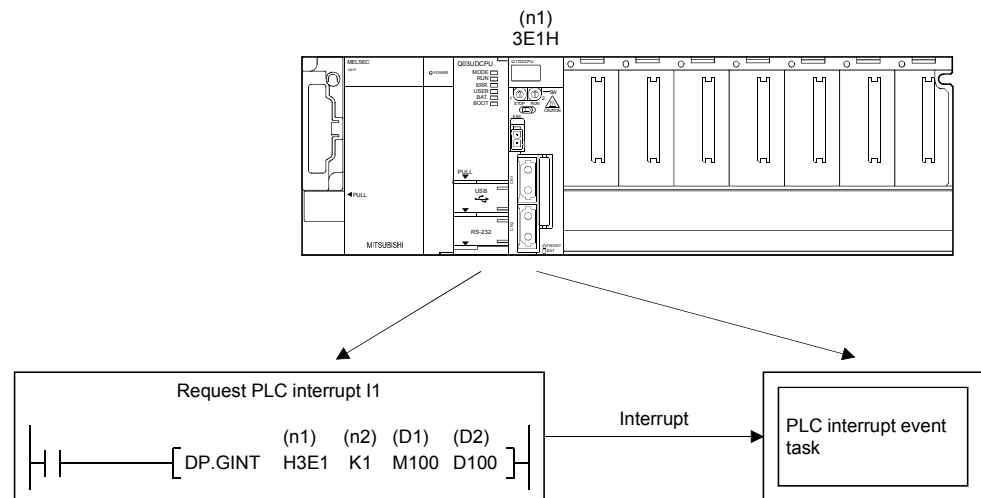
Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. (Note-1) CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(n2)	Interrupt instruction No.	User	16-bit binary
(D1) (Note-1)	Complete devices (D1+0) : Device which make turn on for one scan at accept completion of instruction. (D1+1) : Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) (Note-1)	Complete status storage device	System	Word

(Note-1) : Omission possible with both of (D1) and (D2) omission.

3 MOTION DEDICATED PLC INSTRUCTION

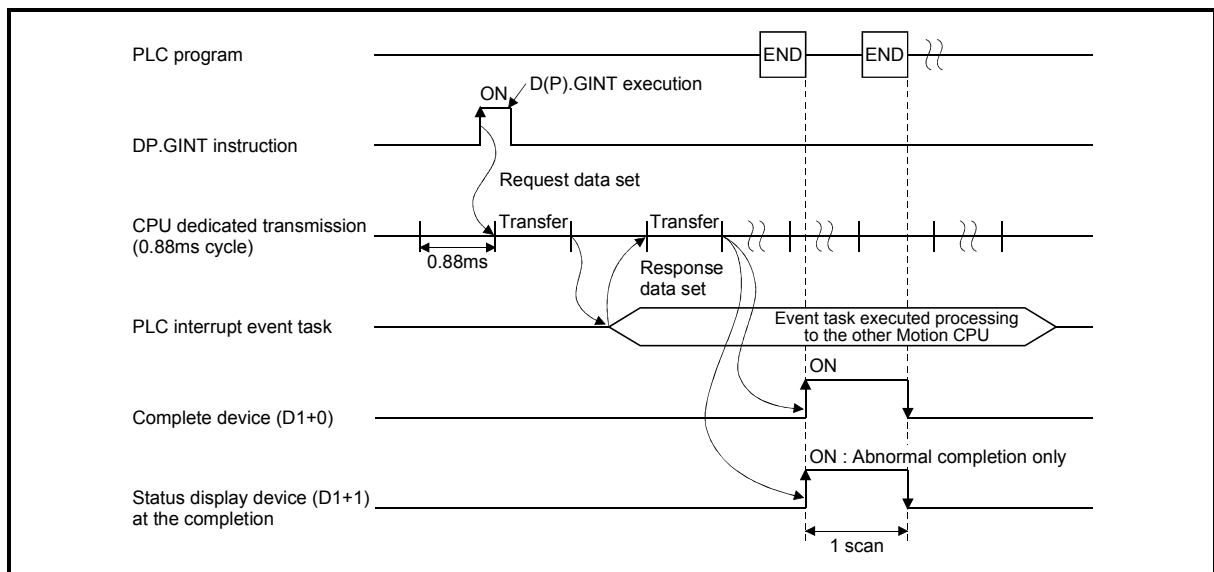
[Controls]

- (1) Processing for the active program (operation program status) of Motion SFC program set in the "PLC interruption of event task" is executed by the execution instruction of D(P).GINT instruction.
- (2) This instruction is always valid regardless of the state of real mode/virtual mode/ mode switching when the operating system software of Motion CPU is SV22.
- (3) Event processing is not executed when the Motion CPU side is DI (interrupt disable). Execute the EI (interrupt enable) instruction before event processing.



[Operation]

Outline operation between CPUs at the DP.GINT instruction execution is shown below.



3 MOTION DEDICATED PLC INSTRUCTION

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).

If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status ^(Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct PLC program.
2082	The interrupt pointer No. set in the D(P).GINT instruction is outside the range of 0 to 15.	
2100	There are 33 or more simultaneous D(P).GINT instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	

(Note) : 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

Error code ^(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct PLC program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	

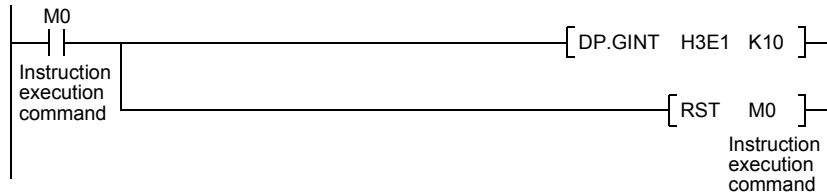
(Note) : 0 (Normal)

3 MOTION DEDICATED PLC INSTRUCTION

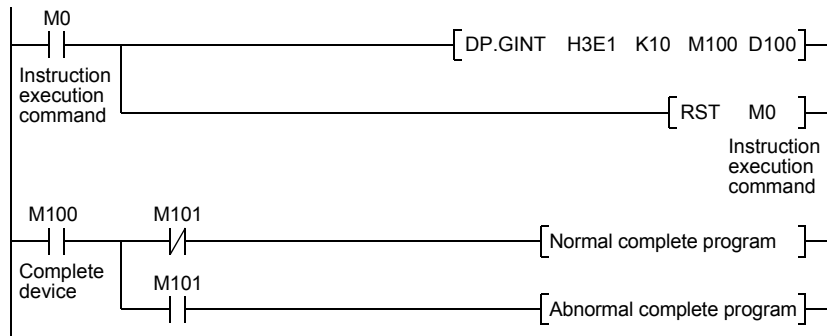
[Program example]

Program which generates interrupt of the interrupt pointer number 10 toward the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omit the complete device and complete status.



<Example 2> Program which use the complete device and complete status.



3 MOTION DEDICATED PLC INSTRUCTION

3.3 Precautions

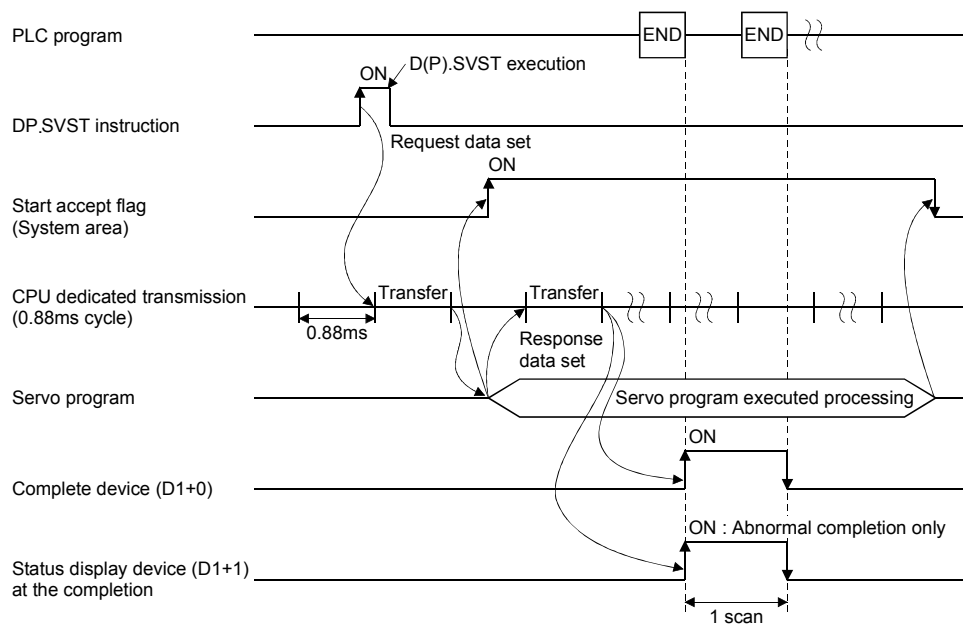
(1) CPU shared memory address used in Motion dedicated instruction

(a) Start accept flag

The status of each flag is stored in the following address.

CPU shared memory address	Description																					
204H(516) 205H(517)	<p>The start accept flag for 32 axes are stored corresponding to each bit. (Set flag: Q173DCPU : J1 to J32/Q172DCPU : J1 to J8)</p> <p>OFF : Start accept flag enable ON : Start accept flag disable</p> <table><tr><td></td><td>b15</td><td>b14</td><td></td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>204H(516) address</td><td>J16</td><td colspan="3">.....</td><td>J2</td><td>J1</td></tr><tr><td>205H(517) address</td><td>J32</td><td colspan="3">.....</td><td>J18</td><td>J17</td></tr></table>		b15	b14		b2	b1	b0	204H(516) address	J16			J2	J1	205H(517) address	J32			J18	J17
	b15	b14		b2	b1	b0																
204H(516) address	J16			J2	J1																
205H(517) address	J32			J18	J17																

The start accept flag is set after instruction acceptance of by the Motion CPU as follows.



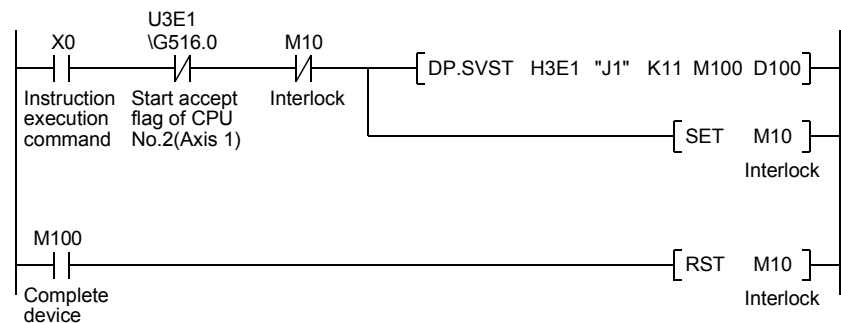
3 MOTION DEDICATED PLC INSTRUCTION

The start accept flag does not turn ON until the instruction accepting of instruction is completed by the Motion CPU after instruction execution by the PLC CPU.

Therefore, use a user device created interlock as required to prevent the execution of the next Motion dedicated PLC instruction and avoid a same axis double start error.

[Program example]

Program which executes continuous start of servo program No.11 for Axis 1 of the Motion CPU (CPU No.2), while X0 is ON.



(b) "Fixed at 0" area

The following area, which is used in Q173HCPU/Q172HCPU/Q173CPU(N)/Q172CPU(N) is not used in Q173DCPU/Q172DCPU and is therefore "Fixed at 0" for these processor.

The following interlocks are not used in new Q173DCPU/Q172DCPU ladder program.

CPU shared memory address (Decimal address)	Description (Q173HCPU/Q172HCPU/Q173CPU(N)/Q172CPU(N))
30H(48)	Fixed at 0 (To self CPU high speed interrupt accept flag from CPU1)
31H(49)	Fixed at 0 (To self CPU high speed interrupt accept flag from CPU2)
32H(50)	Fixed at 0 (To self CPU high speed interrupt accept flag from CPU3)
33H(51)	Fixed at 0 (To self CPU high speed interrupt accept flag from CPU4)
206H(518)	Fixed at 0 (Speed changing flag (Axis1 to 16))
207H(519)	Fixed at 0 (Speed changing flag (Axis17 to 32))
208H(520)	Fixed at 0 (Synchronous encoder current value changing flag (Axis1 to 12))
20CH(524)	Fixed at 0 (Current value within 1cam shaft revolution changing flag (Axis1 to 16))
20DH(525)	Fixed at 0 (Current value within 1cam shaft revolution changing flag (Axis17 to 32))

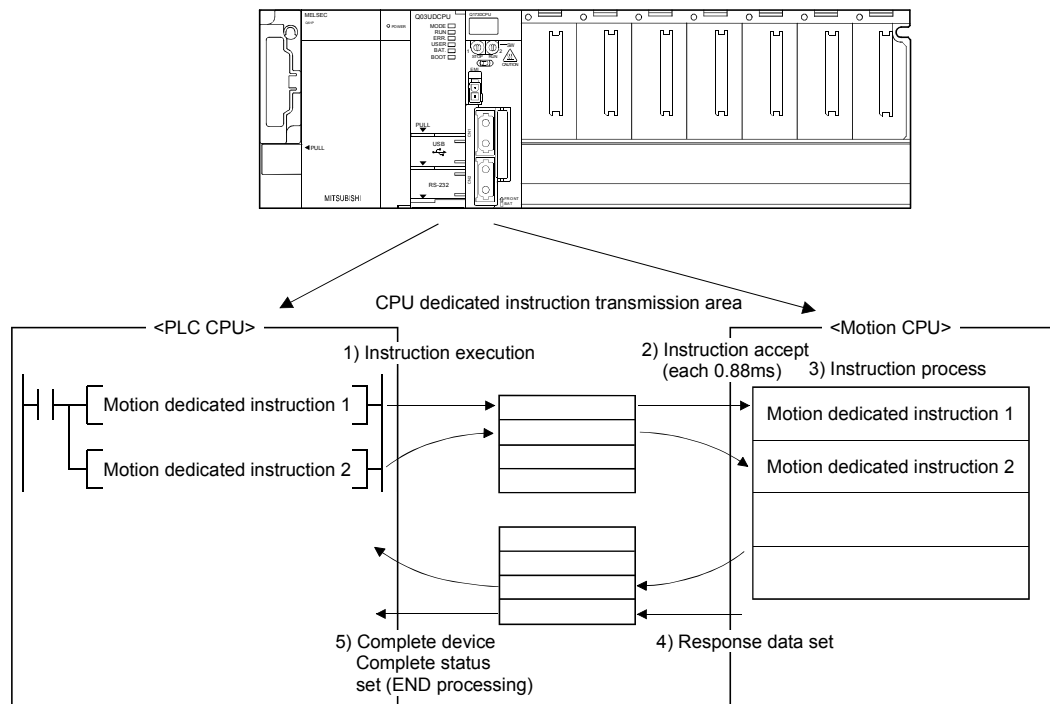
3 MOTION DEDICATED PLC INSTRUCTION

(2) CPU dedicated instruction transmission

(a) Outline operation of Motion Dedicated PLC Instruction

Motion dedicated PLC instruction is transmitted through the CPU dedicated instruction transmission area set up in the system area on shared memory at the Multiple CPU high speed transmission.

Outline operation for Motion dedicated PLC instruction is shown below.



CPU dedicated instruction transmission area shown in table below is allocated as initial setting.

Table 3.1 Number of CPU dedicated instruction transmission area

Number of Multiple CPU modules	Number of CPU dedicated instruction transmission area for each target CPU
2	47 blocks
3	23 blocks
4	15 blocks

3 MOTION DEDICATED PLC INSTRUCTION

As shown in Table 3.2, each Motion dedicated PLC instruction uses a certain number of blocks in the CPU dedicated instruction transmission area until the "complete device" turns on by the PLC CPU after instruction execution.

Table 3.2 Number of blocks used for Motion dedicated PLC instruction

Instructions	Number of blocks used
D(P).SFCS	1
D(P).SVST	1
D(P).CHGA	1
D(P).CHGV	1
D(P).CHGT	1
D(P).DDWR	2 (Note)
D(P).DDRDR	2 (Note)
D(P).GINT	1

(Note) : When the number of transmitted data is 4 words or less, number of blocks used is 1.

[Operation example]

Below is an example when 12 D(P).SVST instructions and 12 D(P).DDWR instructions (5 word or more each) are executed simultaneously.

The number of blocks used is as follows;

$$\begin{aligned} & 12 \text{ D(P).SVST instructions} \times 1 \text{ block each} + \\ & 12 \text{ D(P).DDWR instructions} \times 2 \text{ blocks each} \\ & = 36 \text{ (Total blocks used)} \end{aligned}$$

(b) Permissible number of executions for dedicated instructions on the Multiple CPU high speed bus

When the number of blocks being used to communicate with each CPU in the Multiple CPU dedicated instruction transmission area exceeds the set value for "maximum number of blocks used for dedicated instruction" in the Multiple CPU high speed bus setting (special registers SD796 to SD799 of PLC CPU), the system enters a state where the Motion dedicated PLC instruction is not accepted (permissible number of executions exceeded state). At the time of Motion dedicated instruction execution towards the target CPU on the Multiple CPU high speed bus, an abnormal complete status "0010H" is set in the complete status device. If the complete device is omitted, no operation occurs at all.

An interlock can be created using special relays containing block-use information (SM796 to SM799 of the PLC CPU) so that the permissible number of executions is not exceeded.

3 MOTION DEDICATED PLC INSTRUCTION

• Special register of PLC CPU

Device No.	Name	Detail	Description	Set by
SD796	Multiple CPU high speed bus maximum number of blocks (For CPU No.1)	Maximum number of blocks used for dedicated instruction range 1 to 9 (Default: 2) (Note): When a value other than 1 to 9 is set, operation is performed as if 9 is set.	Specifies the maximum number of blocks used for the dedicated instruction of Multiple CPU high speed bus. When the dedicated instruction of Multiple CPU transmission is executed to the target CPU, and the number of empty blocks of the dedicated instruction transmission area is less than the setting value of this register, the block information using dedicated instruction of Multiple CPU high speed bus (SM796 to SM799) is turned ON, which is used as the interlock signal for consecutive execution of the dedicated instruction of Multiple CPU transmission.	User (At 1 scan after RUN)
SD797	Multiple CPU high speed bus maximum number of blocks (For CPU No.2)			
SD798	Multiple CPU high speed bus maximum number of blocks (For CPU No.3)			
SD799	Multiple CPU high speed bus maximum number of blocks (For CPU No.4)			

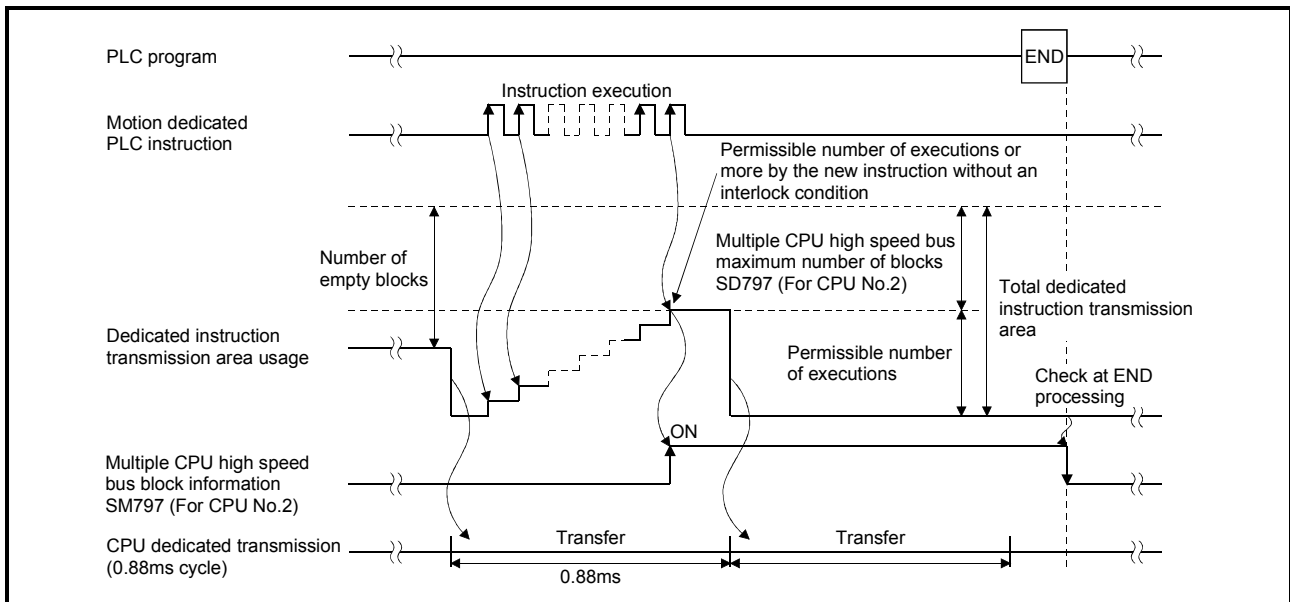
• Special relay of PLC CPU

Device No.	Name	Detail	Description	Set by
SM796	Multiple CPU high speed bus block information (For CPU No.1)	OFF: Block is secured ON : Block set by SD796 cannot be secured	Turns ON when the number of the remaining blocks of the dedicated instruction transmission area used for the dedicated instruction of Multiple CPU high speed bus is less than the number of blocks specified by maximum number of blocks used for dedicated instruction of Multiple CPU high speed bus setting. Turns ON at instruction execution. Turns OFF when empty area exists at END processing.	System (When instruction/END processing executed)
SM797	Multiple CPU high speed bus block information (For CPU No.2)	OFF: Block is secured ON : Block set by SD797 cannot be secured		
SM798	Multiple CPU high speed bus block information (For CPU No.3)	OFF: Block is secured ON : Block set by SD798 cannot be secured		
SM799	Multiple CPU high speed bus block information (For CPU No.4)	OFF: Block is secured ON : Block set by SD799 cannot be secured		

3 MOTION DEDICATED PLC INSTRUCTION

[Operation timing]

Operation which executes each Motion dedicated instruction and turns on the Multiple CPU high speed bus block information.



[Operation example]

When multiple D(P).DDWR instructions (5 word or more each) are executed simultaneously before turning on each complete device in the 2 Multiple CPUs. If the number of blocks used for each item is set as follows,

- Number of CPU dedicated instruction transmission area: 47 blocks (Initial value)
- Multiple CPU high speed bus block information (SD797): 2 (Initial value)
- D(P).DDWR number of blocks used

And, when 23 D(P).DDWR instructions are issued within the Multiple CPU high speed transmission cycle (0.88 ms), the number of blocks used is as follows.

- 23 D(P).DDWR instructions × 2 blocks each = 46 (Total blocks used)

Therefore, the number of empty blocks is as follows;

- 47 (Number of CPU dedicated instruction transmission area) - 46 (Total blocks used) = 1 (Number of empty blocks)

3 MOTION DEDICATED PLC INSTRUCTION

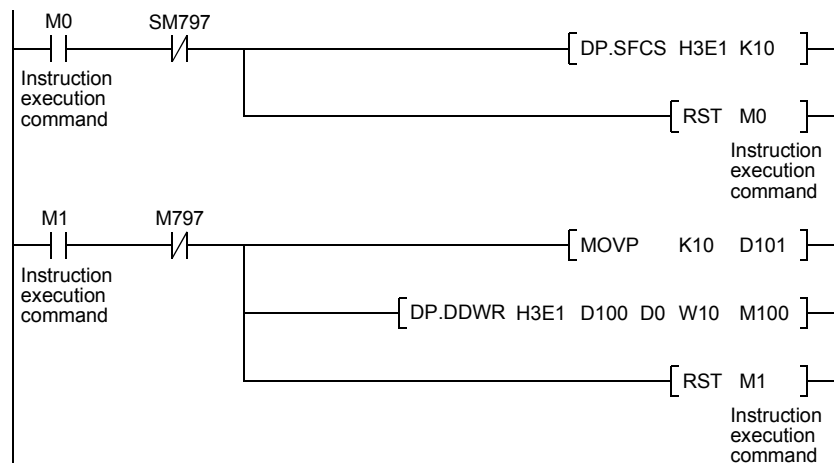
- 1 (Number of empty blocks) < 2 (Multiple CPU high speed bus maximum number of blocks (SD797))

In the above case, the number of empty blocks is less than the Multiple CPU high speed bus maximum number of blocks (SD797), therefore Multiple CPU high speed bus block information (SM7979) turns on.

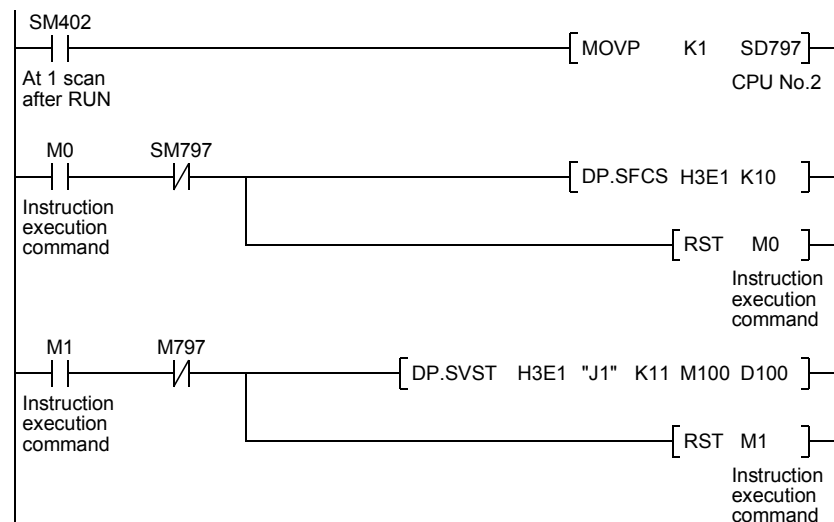
If a new instruction is executed while in this status, it will be more than the permissible number of executions. However, this can be avoided by using SM797 as an interlock.

[Program example]

<Example 1> Program which sets 2 (Initial value) to SD797 and uses SM797 as an interlock when DP.DDWR (Number of blocks used : 2) is executed.



<Example 2> Program which sets 1 to SD797 and uses SM797 as a condition when D(P).DDWR/D(P).DDRD is not executed.



3 MOTION DEDICATED PLC INSTRUCTION

(c) CPU dedicated instruction transmission area

If the size of the CPU dedicated instruction transmission area is insufficient, it can be increased changing the system area size. The size of the CPU dedicated instruction transmission area is decided depending on the number of CPU modules used and selected system area size as follows.

Refer to the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for details of the system area size change.

• Number of Multiple CPU modules : 2

Selected system area size	Number of CPU dedicated instruction transmission area for each target CPU
1.0k word	47 blocks
2.0k word	111 blocks

• Number of Multiple CPU modules : 3

Selected system area size	Number of CPU dedicated instruction transmission area for each target CPU
1.0k word	23 blocks
2.0k word	55 blocks

• Number of Multiple CPU modules : 4

Selected system area size	Number of CPU dedicated instruction transmission area for each target CPU
1.0k word	15 blocks
2.0k word	36 blocks

(d) Number of simultaneous instruction acceptance for Motion CPU

The following number of instructions can be accepted simultaneously in the Motion CPU.

- D(P).SFCS: 64
- Total of D(P).SVST and D(P).CHGA: 64
- D(P).GINT: 32
- Total of D(P).DDRD and D(P).DDWR: 64
- D(P).CHGV/D(P).CHGT: Last instruction executed is valid. There is not a limitation for number of simultaneous instruction acceptance.

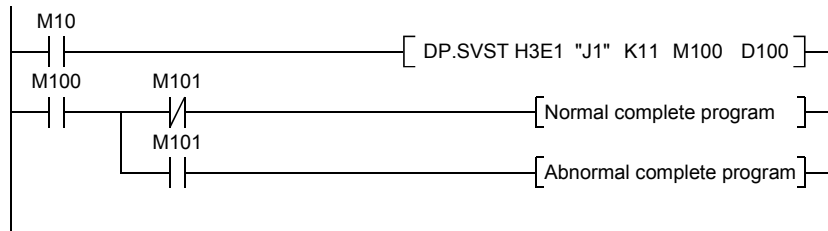
When more than the above number of instructions are executed by the PLC CPU, even if there is enough area in the CPU dedicated instruction transmission area, the Motion CPU cannot accept it.

In this case, 2100 is set to the complete status information and it abnormal completion occurs.

3 MOTION DEDICATED PLC INSTRUCTION

(3) Execution of Motion dedicated PLC instruction

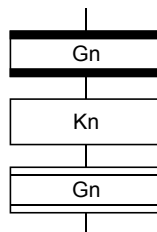
- (a) Motion dedicated PLC instruction can be executed with a fixed cycle execute type program and interrupt program. However, the complete device is a pulse-type. If the complete device (M100 in below example) is set, it may not be recognized during the PLC scan. Therefore, the PLC program should scan for completion of the device and use a set bit to execute the Motion instruction string.



- (b) The below devices cannot be used as program file registers or local devices.

- Each instruction's complete device and complete status
- D1 of D(P).DDRDI instruction (First device of the self CPU where the reading data is stored.)

- (c) When using the Motion dedicated function of the operation control step (Fn/FSn) and Motion control program (Kn) in Motion CPU. Since there is no instruction execution flag in the PLC CPU, it is necessary to create a user-defined interlock using wait transition (Gn) as shown below.



3 MOTION DEDICATED PLC INSTRUCTION

(4) Complete status information

The codes stored in complete status at the completion of Motion dedicated PLC instruction are shown below.

If the complete status storage device is omitted, an error is not detected and operation becomes "No operation".

Complete status (Error code) (H)	Error factor
0	Normal completion
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value. (Permissible value is different depending on the number of CPU modules.).
2000 (Note)	Command that cannot be decoded in the Motion CPU was specified.
2001 (Note)	The specified device cannot be used in the Motion CPU, or it is outside the device range.
2002 (Note)	A Motion dedicated PLC instruction that does not correspond with the operating system of the Motion CPU was executed.
2080 (Note)	Number of writing data points set by D(P).DDWR instruction is wrong.
2081 (Note)	Number of reading data points set by D(P).DDRDR instruction is wrong.
2082 (Note)	The interrupt pointer No. set in the D(P).GINT instruction is outside the range of 0 to 15.
2100 (Note)	<ul style="list-style-type: none"> • D(P).SFCS instruction use There are 65 or more simultaneous D(P).SFCS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • D(P).SVST/D(P).CHGA instruction use There are 65 or more simultaneous D(P).SVST/D(P).CHGA sum table instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • D(P).GINT instruction use There are 33 or more simultaneous D(P).GINT instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • D(P).DDRDR/D(P).DDWR instruction use There are 65 or more simultaneous D(P).DDRDR/D(P).DDWR sum table instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.
2200 (Note)	The starting Motion SFC program No. is outside the range of 0 to 255.
2201 (Note)	The servo program No. to execute is outside the range of 0 to 4095.
2202 (Note)	Axis No. set by D(P).SVST instruction is wrong.
2203 (Note)	Axis No. set by D(P).CHGA instruction is wrong.
2204 (Note)	Axis No. set by D(P).CHGV instruction is wrong.
2205 (Note)	Axis No. set by D(P).CHGT instruction is wrong.

(Note) : The error code is dedicated with the Motion CPU.

3 MOTION DEDICATED PLC INSTRUCTION

(5) Order of instruction execution

Methods to control using execution data after it is transmitted from the PLC CPU to the Motion CPU are shown below.

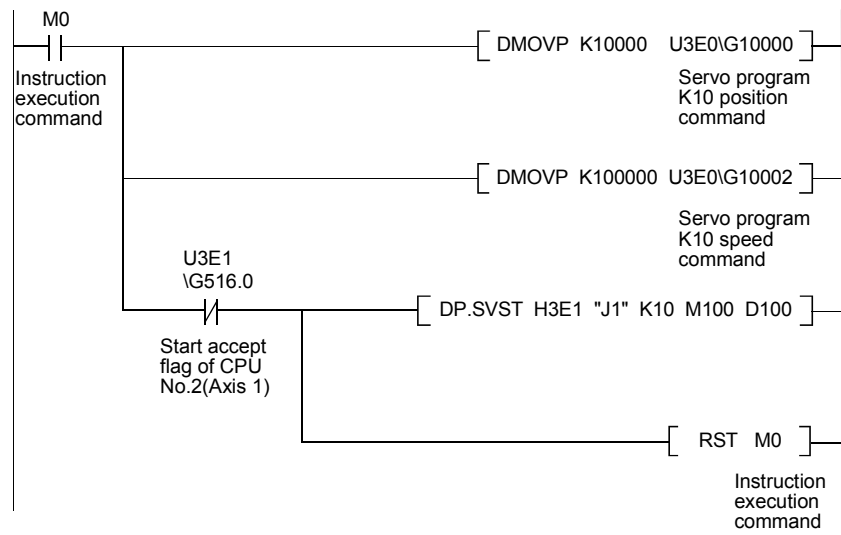
- (a) Method to execute after data is written to the shared memory area (Multiple CPU high speed transmission area).

Write the data from PLC CPU to the shared memory area (Multiple CPU high speed transmission area) of the self CPU, and then it can be utilized for Motion dedicated PLC instruction execution.

[Program example]

Program which starts the servo program (positioning) by DP.SVST instruction after the data has been writing to shared memory area (Multiple CPU high speed transmission area (U3E0\G10000 to U3E0\G10003) from PLC CPU (CPU No.1).

Ladder (PLC CPU side)



Servo program (Motion CPU side)

[K10: REAL]
1 INC-1
Axis 1, U3E0\G10000 μ m
Speed U3E0\G10002 mm/min

3 MOTION DEDICATED PLC INSTRUCTION

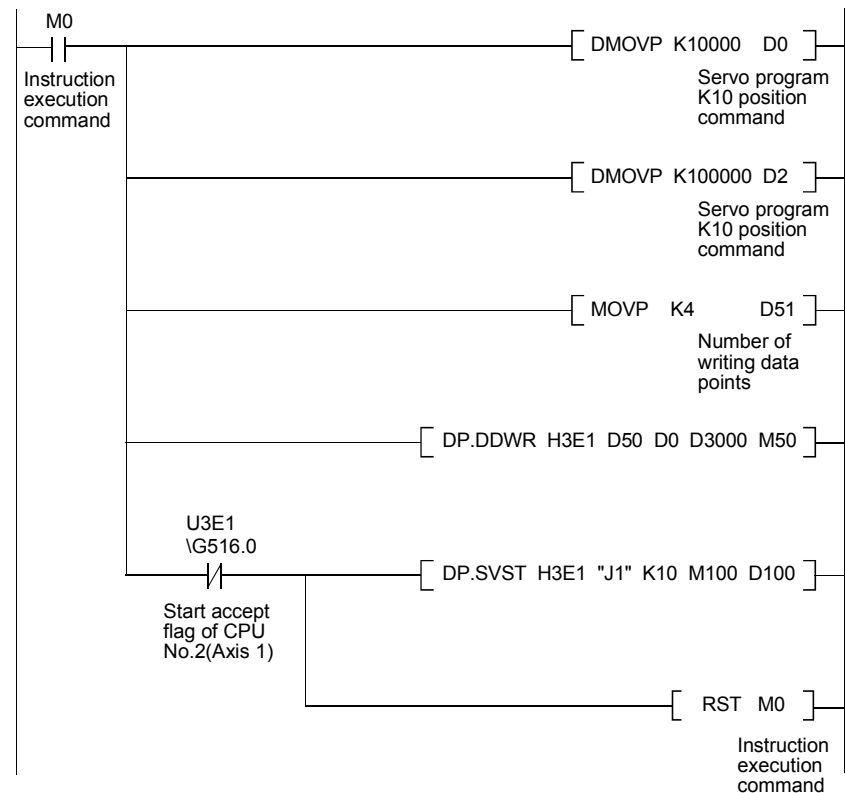
(b) Method to execute after data is written by D(P).DDWR instruction

Write the data from the PLC CPU to the Motion CPU by D(P).DDWR instruction, and then it can be utilized for Motion dedicated PLC instruction execution.

[Program example]

Program which starts the servo program (positioning) by DP.SVST instruction after data is written to D3000 to D3002 of the Motion CPU (CPU No.2) from the PLC CPU (CPU No.1) by DP.DDWR.

Ladder (PLC CPU side)



Servo program (Motion CPU side)

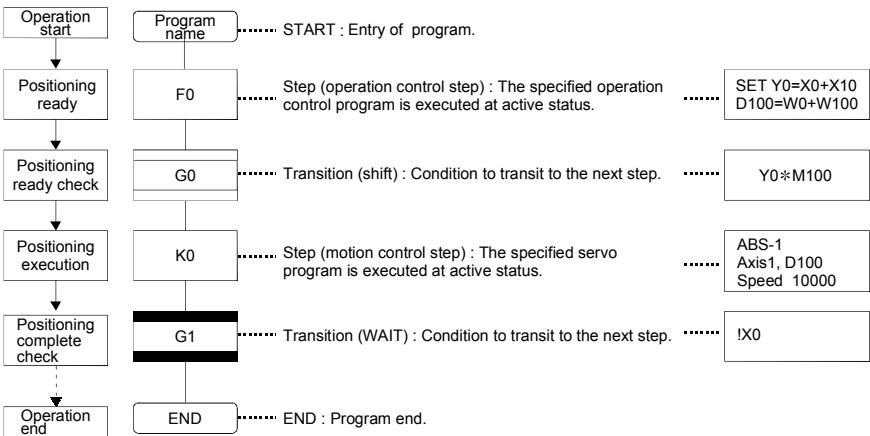
[K10: REAL]
1	INC-1	
Axis	1,	D3000 μ m
Speed		D3002 mm/min

4. MOTION SFC PROGRAMS

Refer to Chapter "12 ERROR CODE LISTS" for details of Motion SFC program error.

4.1 Motion SFC Program Configuration

The Motion SFC Program is constituted by the combination of start, steps, transitions, end and others are shows below.



The above Motion SFC program to be started performs the following operations.

- (1) The step (F0) is activated and the operation specified with the step (F0) is executed (positioning ready). A step in such an active state is called an active step.
- (2) Whether the condition specified with the transition (G0) has enabled or not (whether the positioning program can be started or not) is checked. The active step (F0) is deactivated at the completion of condition and the next step (K0) is activated (servo program (K0) is started).
- (3) The operating completion of the step (K0) (positioning completion of the servo program K0) is checked, and control transits to the next step at operating completion (completion of condition).
- (4) With the transition of the active step as described in above (1) to (3), control is executed and ends at END.

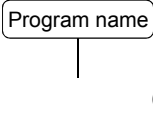
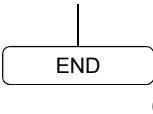
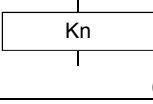
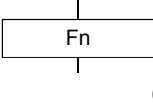
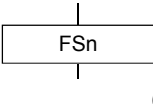
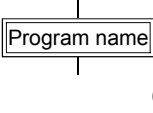
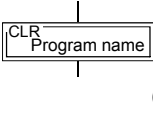
Refer to Section "9.2.2 Task operation" for details of the execution timing of the Motion SFC program such as above.

POINT
The number of steps which can be active steps simultaneously is up to 256, with those of all Motion SFC programs combined. Excess of 256 will result in the Motion SFC Program error 16120.
Each symbol of the Motion SFC program is as follows.
F/FS : Operation control, K : Positioning control, G : Judgment

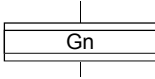
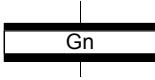
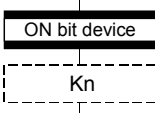
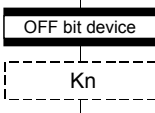
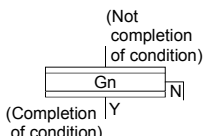
4 MOTION SFC PROGRAMS

4.2 Motion SFC Chart Symbol List

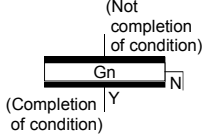
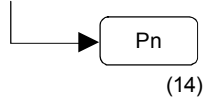
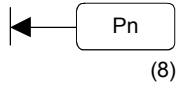
Parts as Motion SFC program components are shown below. The operation sequence or transition control is expressed with connecting these parts by directed lines in the Motion SFC program.

Classification	Name	Symbol (Code size (byte))	List Representation	Function
Program start/end	START		Program name	<ul style="list-style-type: none"> Indicates an entry of program as a program name. Specify this program name at a subroutine call. Only one program name for one program.
	END		END	<ul style="list-style-type: none"> Indicates an end (exit) of program. When a subroutine call was carried out, returns to the call source program. Multiple program names or no symbols for one program.
Step	Motion control step		CALL Kn	<ul style="list-style-type: none"> Starts a servo program Kn (K0 to K4095).
	Once execution type operation control step		CALL Fn	<ul style="list-style-type: none"> Execute once the operation control program Fn (F0 to F4095).
	Scan execution type operation control step		CALL FSn	<ul style="list-style-type: none"> Repeats an operation control program FSn (FS0 to FS4095) until the next transition condition enables.
	Subroutine call/start step		GSUB program name	<ul style="list-style-type: none"> When the next of GSUB is WAIT, performs "subroutine call" and transits control to the specified program. Control returns to the call source program at END execution. When the next of GSUB is except WAIT, performs "subroutine start", and starts the specified program and transits to the next (lower part). The start source and destination programs are executed simultaneously, and the call destination program ends at END execution.
	Clear step		CLR program name	<ul style="list-style-type: none"> Stops and ends the specified program running. After an end, it is started from the initial (start step) by restarting the program. When the specified program is during "subroutine call", the subroutine program is also stopped to execute. When the specified program is after "subroutine start", the subroutine program is not stopped to execute. When clearing to the subroutine by which the "subroutine call" was executed, the specified subroutine is stopped to execute, returns to the call source program, and transits to the next.

4 MOTION SFC PROGRAMS

Classification	Name	Symbol (Code size (byte))	List representation	Function
Transition	Shift (Pre-read transition)	 (8)	SFT Gn	<ul style="list-style-type: none"> When just before is the motion control step, transits to the next step by formation of transition condition Gn (G0 to G4095) without waiting for the motion operating completion. When just before is the operation control step, transits to the next step by the completion of transition condition after operating execution. When just before is subroutine call or starting step, transits to the next step by formation of transition condition without waiting for the operating completion of subroutine.
	WAIT	 (8)	WAIT Gn	<ul style="list-style-type: none"> When just before is the motion control step, waits for the motion operating completion and then transits to the next step by the completion of transition condition Gn (G0 to G4095). When just before is the operation control step, transits to the next step by formation of transition condition after operating execution. (Same operation as Shift.) When just before is subroutine call or starting step, waits for the operating completion of subroutine and then transits to the next step by the completion of transition condition.
	WAITON	 (14)	WAITON bit device	<ul style="list-style-type: none"> Prepares for starting of the next motion control step, and issues an instruction immediately when the specified bit device turns ON. Always pair this transition with the motion control step one-for-one.
	WAITOFF	 (14)	WAITOFF bit device	<ul style="list-style-type: none"> Prepares for starting of the next motion control step, and issues an instruction immediately when the specified bit device turns OFF. Always pair this transition with the motion control step one-for-one.
	Shift Y/N	 (Not completion of condition) (Completion of condition)	IFBm IFT1 SFT Gn : JMP IFEm IFT2 SFT Gn+? : JMP IFEm IFEm	<ul style="list-style-type: none"> When just before is the motion control step, transits to the next step by formation of transition condition Gn (G0 to G4095) without waiting for the motion operating completion. If not formation of transition condition, transits to the right-connected step. When just before is the operation control step, transits to the next step by the completion of transition condition after operating execution. If not the completion of transition condition, transits to the right-connected step. When just before is "subroutine call" or "starting step", transits to the next step by the completion of transition condition without waiting for the operating of subroutine completion. If not formation of transition condition, transits to the right-connected step.

4 MOTION SFC PROGRAMS

Classification	Name	Symbol (Code size (byte))	List representation	Function
Transition	WAIT Y/N		IFBm IFT1 WAIT Gn : JMP IFEm IFT2 WAIT Gn+? : JMP IFEm IFEm	<ul style="list-style-type: none"> When just before is the motion control step, waits for the motion operating completion and then transits to the next step by formation of transition condition Gn (G0 to G4095). If not completion of transition condition, transits to the right-connected step. When just before is the operation control step, transits to the next step by the completion of transition condition after operating execution. If not the completion of transition condition, transits to the right-connected step. (Same operation as Shift.) When just before is subroutine call or starting step, waits for the operating completion of subroutine, and then transits to the next step by the completion of transition condition. If not formation of transition condition, transits to the right-connected step.
Jump	Jump		JMP Pn	<ul style="list-style-type: none"> Jumps to the specified pointer Pn (P0 to P16383) of the self program.
Pointer	Pointer		Pn	<ul style="list-style-type: none"> Indicates a jump destination pointer (label). This pointer can be set at a step, transition, branch point or coupling point. P0 to P16383 can be set in one program. The same No. may also be used in other programs.

4 MOTION SFC PROGRAMS

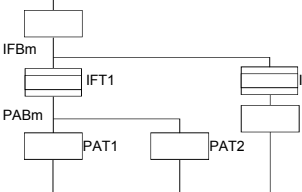
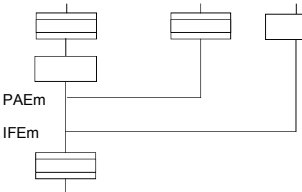
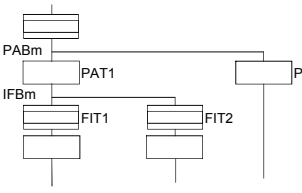
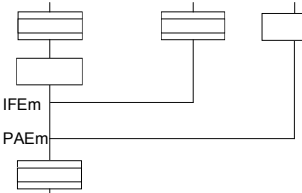
4.3 Branch and Coupling Chart List

Branch and coupling patterns which specify step and transition sequences in the Motion SFC charts are shown below.

	Name (Code size (byte))	Motion SFC chart symbol	List representation	Function
Basic type	Series transition (Corresponding symbol size)		List representation corresponding to the Motion SFC chart symbols shown in Section 4.2.	<ul style="list-style-type: none"> Steps and transitions connected in series are processed in order from top to bottom. Steps and transitions need not be lined up alternately. When a transition is omitted, unconditional shift processing is performed.
	Selective branch ((Number of branches + 2) × 10)		CALL Kn IFBm IFT1 SFT Gn CALL Fn : JMP IFE _m	<ul style="list-style-type: none"> The route which transition condition enables first is executed after executing the step or transition preceding a branch. Selective branch destinations should always be started by transitions, all of which must be Shift or WAIT. (Using Shift and WAIT together will cause a parallel branch.)
	Selective coupling (8)		IFT2 SFT Gn' CALL Fn' : (JMP IFE _m) IFE _m CALL Fn''	<ul style="list-style-type: none"> After the route branched by a selective branch has been processed, execution shifts to a coupling point. A coupling may be preceded and followed by either a step or a transition.
	Parallel branch (Number of branches × 22 + number of coupling points × 2 + 12)		CALL Kn PABm PAT1 CALL Fn SFT Gn' : JMP PAE _m	<ul style="list-style-type: none"> Multiple routes (steps) connected in parallel are executed simultaneously. Each parallel branch destination may be started by either a step or transition.
	Parallel coupling (8)		PAT2 CALL Fn' SFT Gn'' : (JMP PAE _m) PAE _m CALL Fn'' :	<ul style="list-style-type: none"> Execution waits at the coupling point for executions of the routes branched by a parallel branch to be completed, and shifts to the next when executions of all routes are completed. A coupling may be preceded and followed by either a step or a transition. When this coupling is preceded by an FS step, scans are executed during waiting. After waiting is complete, scans are not executed.
	Jump transition (Corresponding symbol size)	<div><Normal jump></div>	CALL Fn JMP Pn	1) Normal jump <ul style="list-style-type: none"> After the step or transition preceding this jump transition is executed, execution shifts to the pointer Pn specified within its own program. The jump destination may either be a step or transition. When a jump takes place from an FS step to a transition, scans are executed during waiting for the completion of transition condition of the jump destination.
		<div><Coupling jump></div>		
			Pn CALL Fn' CALL Kn	2) Coupling jump <ul style="list-style-type: none"> When a jump to the other route within a parallel branch takes place after the parallel branch, a "coupling jump" takes place and execution waits at the jump destination.

4 MOTION SFC PROGRAMS

Combining the basic type branches/couplings provides the following application types, which are defined as in the basic types.

	Name	Motion SFC chart symbol	List representation	Function
Applica- tion type	Selective branch Parallel branch		CALL Kn IFBm IFT1 SFT Gn PABm PAT1 CALL Fn : JMP PAEm PAT2 CALL Fn' : (JMP PAEm) PAEm JMP IFEEm IFT2 SFT Gn' CALL Fn'' : (JMP IFEEm) IFEEm SFT Gn''	<ul style="list-style-type: none"> After a selective branch, a parallel branch can be performed.
	Parallel coupling Selective coupling		IFT2 SFT Gn' CALL Fn'' : (JMP IFEEm) IFEEm SFT Gn''	<ul style="list-style-type: none"> The selective coupling point can be the same as the coupling point of a parallel coupling for selective branch → parallel branch. Note that in the Motion SFC chart, this type is displayed in order of a parallel coupling → a selective coupling, as shown on the left. In this case, a pointer (Pn) cannot be set between the parallel coupling point (PAEm) and the selective coupling point (IFEEm).
	Parallel branch Selective branch		SFT Gn PABm PAT1 PAT2 CALL Fn IFBm IFT1	<ul style="list-style-type: none"> After a parallel branch, a selective branch can be performed.
	Selective coupling Parallel coupling		SFT Gn' CALL Fn' : JMP IFEEm IFT2 SFT Gn'' CALL Fn'' : (JMP IFEEm) IFEEm JMP PAEm PAT2 CALL Fn''' : CALL Kn (JMP PAEm) PAEm SFT Gn'''	<ul style="list-style-type: none"> The parallel coupling point can be the same as the coupling point of a selective coupling for parallel branch → selective branch. Note that in the Motion SFC chart, this type is displayed in order of a selective coupling → a parallel coupling, as shown on the left. In this case, a pointer (Pn) cannot be set between the selective coupling point (IFEEm) and the parallel coupling point (PAEm).

Application type	Application	Application type	Application
Application type	Application	Application type	Application

4 MOTION SFC PROGRAMS

	Name	Motion SFC chart symbol	List representation	Function
Appli- cation type	Selective coupling Parallel branch		: (JMP IFE m) IFE m PAB m PAT 1 CALL Fn : JMP PAEm PAT 2 CALL Fn' : (JMP PAEm) PAEm :	<ul style="list-style-type: none"> The selective coupling point and parallel branch point can be the same. Note that in the Motion SFC chart, this type is displayed in order of a selective coupling → a parallel branch, as shown on the left. In this case, a pointer (Pn) cannot be set between the selective coupling point (IFE m) and the parallel branch point (PAB m).
	Parallel coupling Selective branch		: JMP PAEm PAEm IFB m IFT 1 SFT Gn : JMP IFE m IFT 2 SFT Gn' : (JMP IFE m) IFE m :	<ul style="list-style-type: none"> The parallel coupling point and selective branch point can be the same. Note that in the Motion SFC chart, this type is displayed in order of a parallel coupling → a selective branch, as shown on the left. Execution waits at the parallel coupling point and shifts to the selective branch. In this case, a pointer (Pn) cannot be set between the parallel coupling point (PAEm) and the selective branch point (IFB m).
	Selective coupling Selective branch		: (JMP IFE m) IFE m IFB m+1 IFT 1 SFT Gn : JMP IFE m+1 IFT 2 SFT Gn' : (JMP IFE m+1) IFE m+1 :	<ul style="list-style-type: none"> The selective coupling point and selective branch point can be the same. Note that in the Motion SFC chart, this type is displayed in order of a selective coupling → a selective branch, as shown on the left. In this case, a pointer (Pn) cannot be set between the selective coupling point (IFE m) and the selective branch point (IFB m+1).
	Parallel coupling Parallel branch		: (JMP PAEm) PAEm PAB m+1 PAT 1 CALL Fn : JMP PAEm+1 PAT 2 CALL Fn' : (JMP PAEm+1) PAEm+1 :	<ul style="list-style-type: none"> The parallel coupling point and parallel branch point can be the same. Note that in the Motion SFC chart, this type is displayed in order of a parallel coupling → a parallel branch, as shown on the left. Execution waits at the parallel coupling point and shifts to the parallel branch. In this case, a pointer (Pn) cannot be set between the parallel coupling point (PAEm) and the parallel branch point (PAB m+1).

4 MOTION SFC PROGRAMS

4.4 Motion SFC Program Name

Set the "Motion SFC program name" to the Motion SFC program No.0 to No.255 individually. (Make this setting in the "Motion SFC program management window" on the Motion SFC program edit screen.)

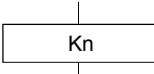
Set the Motion SFC program name within 16 characters. Specify this Motion SFC program name for a "subroutine call/start step (GSUB)" and "clear step (CLR)".

POINT
(1) It is can be set the Motion SFC program to any of No.0 to No.255. There are no specific programs which have special roles.
(2) "\$" cannot be used in the first character of the Motion SFC program name.
(3) "/" : ; , . * ? " < > " cannot be used in Motion SFC program name.

4 MOTION SFC PROGRAMS

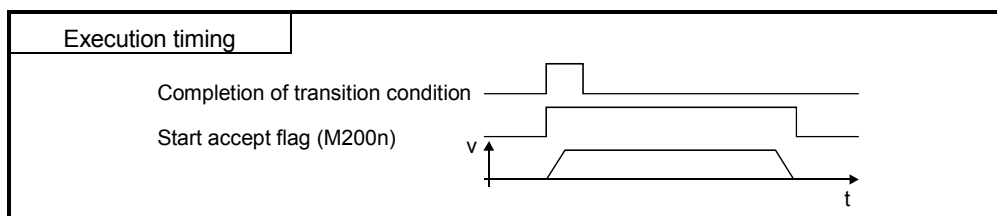
4.5 Steps

4.5.1 Motion control step

Name	Symbol	Function
Motion control step		Starts the servo program Kn. Specified range: K0 to K4095

[Operations]

- (1) Turns on the start accept flag of the axis specified with the specified servo program Kn (n = 0 to 4095) running.
- (2) Starts the specified servo program Kn (n = 0 to 4095).



[Errors]

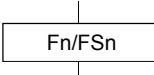
- (1) When the specified servo program Kn does not exist, the Motion SFC program error [16200] will occur and stops to execute the Motion SFC program at the error detection.

[Instructions]

- (1) When the current value change is executed in the Motion SFC program running, specify the CHGA instruction in the servo program and call it at the motion control step.
- (2) If the servo program has stopped due to a major/minor error which occurred at or during a start of the servo program specified with the motion control step, the Motion SFC program continues executing. When the Motion SFC program is stopped at error detection, provide an error detection condition at the transition (transition condition).

4 MOTION SFC PROGRAMS

4.5.2 Operation control step

Name	Symbol	Function
Operation control step		Executes the operation control program Fn/FSn. Specified range: F0 to F4095/FS0 to FS4095

[Operations]

- (1) Once execution type operation control step Fn
In the case of Fn, executes the specified operation control program Fn (n = 0 to 4095) once.
- (2) Scan execution type operation control step FSn
In the case of FSn, repeats the specified operation control program FSn (n = 0 to 4095) until the next transition condition enables.

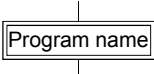
[Errors]

- (1) When the specified operation control program Fn/FSn does not exist, the Motion SFC program error [16201] will occur and stops to execute the Motion SFC program at the error detection.

[Instructions]

- (1) Refer to Chapter "5 OPERATION CONTROL PROGRAMS" for operation expressions that may be described in operation control programs.
- (2) If an operation or similar error occurs the operation control program running, the Motion SFC program continues executing.

4.5.3 Subroutine call/start step

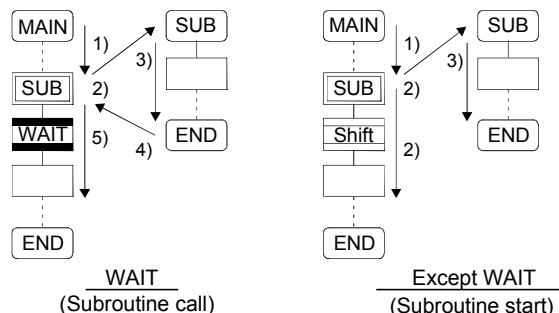
Name	Symbol	Function
Subroutine call/start step		Calls/starts the Motion SFC program of the specified program name.

[Operations]

- (1) Calls/starts the Motion SFC program of the specified program name.
- (2) Control varies with the type of the transition coupled next to the subroutine call/start step.
 - (a) WAIT (Subroutine Call)

When the subroutine call step is executed, control transits to the specified program as shown below, and when END of the called program is executed, control returns to the call source program.
 - (b) Except WAIT (Subroutine Start)

When the subroutine start step is executed, control starts the specified program and then shifts to the next as shown below. Since, the start source and destination Motion SFC programs are executed in parallel. The started program ends at END execution.



[Errors]

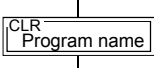
- (1) When the specified Motion SFC program does not exist at a subroutine call/start, the Motion SFC program error [16005] will occur and stops to execute the Motion SFC program at the error detection.
- (2) When the called/started Motion SFC program is already starting at a subroutine call/start, the Motion SFC program error [16006] will occur and stops to execute the Motion SFC program at the error detection.
- (3) When the self program is started at a subroutine call/start, the Motion SFC program error [16110] will occur and stops to execute the Motion SFC program at the error detection.
- (4) When the subroutine to be called/started at a subroutine call/start in the Motion SFC program 2 running which was called/started from the Motion SFC program 1 is the Motion SFC program 1 (call source/start program), the Motion SFC program error [16111] will occur and the call/start source Motion SFC program 2 running is stopped at the point of error detection.

[Instructions]

- (1) There are no restrictions on the depth of subroutine call/start nesting.
- (2) For a subroutine start, the start source Motion SFC program continues processing if the start destination Motion SFC program stops due to an error.
- (3) For a subroutine call, the call source Motion SFC program stops running as soon as the call destination Motion SFC program stops due to an error.

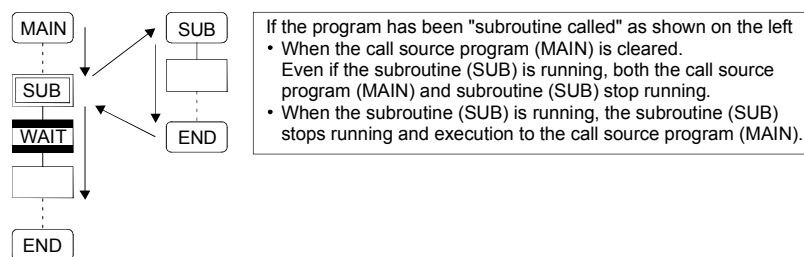
4 MOTION SFC PROGRAMS

4.5.4 Clear step

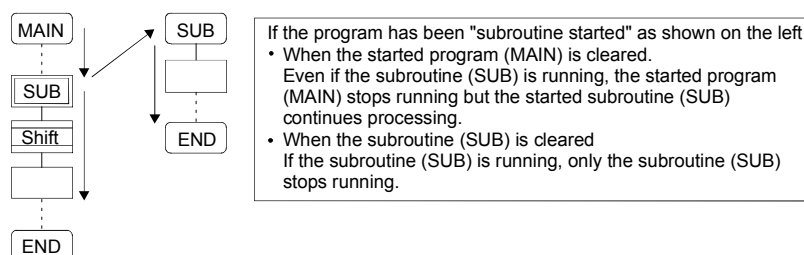
Name	Symbol	Function
Clear step		Stops the Motion SFC program of the specified program name.

[Operations]

- (1) Stops the specified Motion SFC program running.
- (2) The clear-specified Motion SFC program will not start automatically after stopped if it has been set to start automatically.
- (3) The specified program may be its self program.
- (4) If the specified program is being subroutine called, the subroutine program called is also stopped. (Shown below)



- (5) When the specified program has been subroutine started, the subroutine program started continues processing. (Shown below)



- (6) When the servo program started from the specified program is starting, the servo program continues processing.

[Errors]

- (1) When the Motion SFC program specified with the clear step does not exist, the Motion SFC program error [16203] will occur.

[Instructions]

- (1) When the Motion SFC program specified with the clear step is not starting, an error does not occur specifically and this step is ignored.
- (2) If the Motion SFC program running is stopped by the clear step, the output is held.

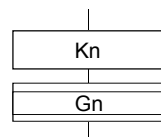
4.6 Transitions

You can describe conditional and operation expressions at transitions. The operation expression described here is repeated until the transition condition enables, as at the scan execution type operation step.

Refer to Chapter "6 TRANSITION PROGRAMS" for the conditional/operation expressions that can be described in transition conditions.

(1) Combinations with motion control steps

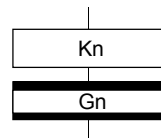
(a) Motion control step + Shift



[Operations]

- Transits to the next step by formation of transition condition Gn without waiting for the operating completion of the servo program Kn started at the motion control step.

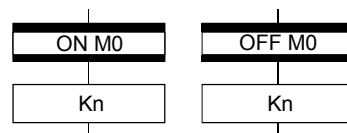
(b) Motion control step + WAIT



[Operations]

- Waits for the operating completion of the servo program Kn started at the motion control step, and then transits to the next step by formation of transition condition Gn.
- The operation completion condition of the servo program Kn is not needed in the transition condition Gn.
- An error stop of the started servo program Kn at/during a start is also regarded as an operation completion.

(c) WAITON/WAITOFF + Motion control step



[Operations]

- Prepares for the start of the motion control step next to WAITON/WAITOFF, and makes a start immediately when the specified bit device turns ON/OFF. When the motion control step is executed without being used with WAITON/WAITOFF, preparations for a start are made after the transition condition preceding the motion control step enables. This will cause a variation of delay/starting time between when the transition condition is completed and when a start is made, but a combination with WAITON/WAITOFF can eliminate the variation of the above delay/starting time.

- Specifiable bit devices

Device	Range
X	X0 to X1FFF
Y	Y0 to Y1FFF
M	M0 to M12287
U□\G	$U□\G10000.0$ to $U□\G(10000+p-1).F$ ^(Note-1) □: CPU No. (No.1: 3E0, No.2: 3E1, No.3: 3E2, No.4: 3E3) [CPU No. that is larger than the number of Multiple CPU cannot be set.]
L	L0 to L8191
B	B0 to B1FFF
F	F0 to F2047

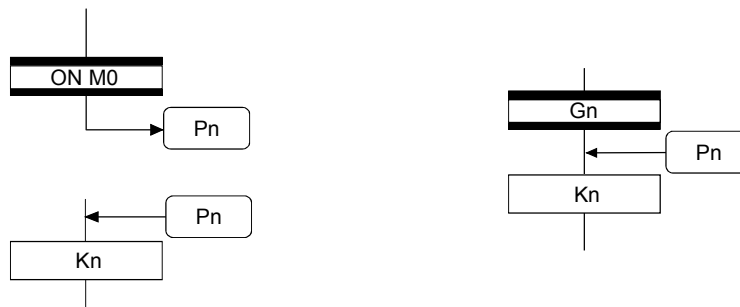
(Note-1): "p" indicates the user setting area points of the Multiple CPU high speed transmission area for the each CPU.

POINT

<p>Refer to Chapter 2 of the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for the user setting area points of the Multiple CPU high speed transmission area.</p>

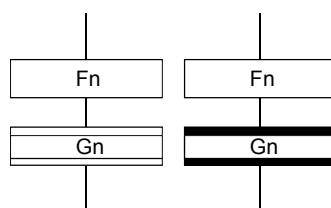
[Instructions]

- Always pair a transition with a motion control step one-for-one. If the step following WAITON/WAITOFF is not a motion control step, the Motion SFC program error [16102] will occur and the Motion SFC program running will stop at the error detection.
- An error will not occur if the jump destination immediately after WAITON/WAITOFF is a motion control step. (Left below)
- A pointer may exist immediately after WAITON/WAITOFF. (Right below)



- If the servo program specified with a motion control step could not be started due to a major/minor error, the Motion SFC program continues running and execution shifts to the next, independently of the WAITON/WAITOFF bit device status. To stop the Motion SFC program at error detection, provide an error detection condition at the next transition (transition condition).
- The following instructions can be used in the motion control step used combining the WAITON/WAITOFF.
(Linear interpolation control, circular interpolation control, helical interpolation, speed switching control, position follow-up control, constant-speed control, high speed oscillation and speed control with fixed position stop.)

(2) Combination with operation control step



[Operations]

- At an operation control step, both Shift and WAIT perform the same operation, and after executing of the operation control program Fn, transits to the next step by formation of transition condition Gn.

(3) Combination with subroutine call/start step

Refer to Section "4.5.3 Subroutine call/start step".

4 MOTION SFC PROGRAMS

4.7 Jump, Pointer

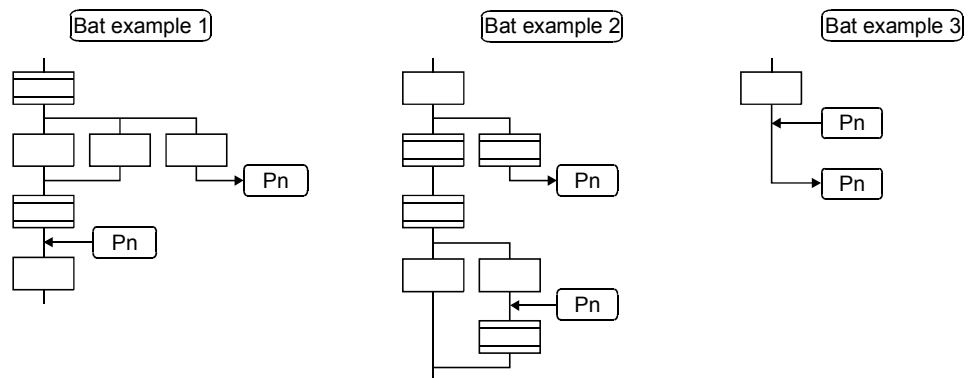


[Operations]

- Setting a jump will cause a jump to the specified pointer Pn of the self program.
- You can set pointers at steps, transitions, branch points and coupling points.
- You can set pointers Pn at P0 to P16383 in one program.

[Instructions]

- You cannot make a jump setting which will exit from within parallel branch-parallel coupling. Connect directly. (Bad example 1 given below)
- You cannot make a jump setting from outside parallel branch-parallel coupling to within parallel branch-parallel coupling. (Bad example 2 given below)
- You cannot make a setting where a label and a jump will continue. (Bad example 3 given below)



4.8 END



[Operations]

- Ends a program. (In this case of an event task or NMI task, operation changes with end operation setting of the program parameter. Refer to Section "9.12 Program Parameters" for details.)
- Making a subroutine call will return to the call source Motion SFC program.

[Instructions]

- END may be set a multiple number of times in one program.
- END cannot be set between a parallel branch and a parallel coupling.
- The output is held after the Motion SFC program is ended by END.

4.9 Branches, Couplings

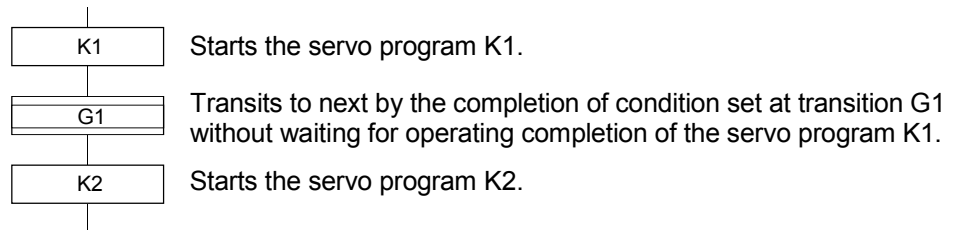
4.9.1 Series transition

Transits execution to the subsequent step or transition connected in series.

- (1) To start a servo program or subroutine and shift execution to the next without waiting for operation completion

Set Shift at a transition.

In this case, the transition (shift) may be omitted. When you omitted the transition, an unconditional shift transition is performed.

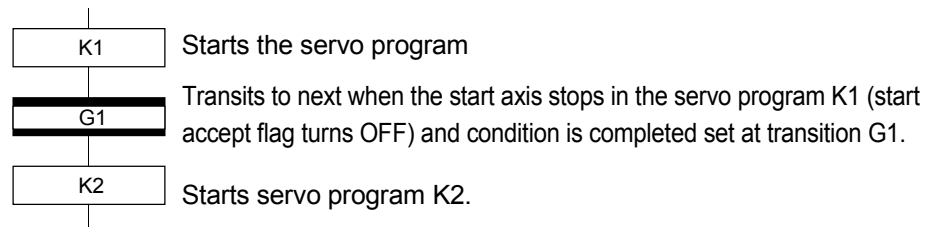


POINT

For a subroutine start, self program and a subroutine program are processed in parallel.

- (2) To start a servo program or subroutine and proceed to the next step on operation completion

Set WAIT at a transition.



POINT

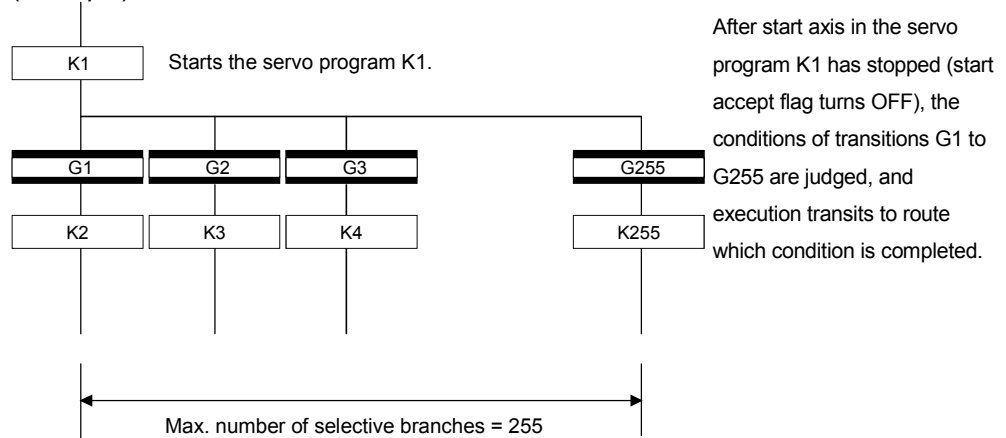
- (1) The above start accept flag of the axis started in the next servo program K2 is not included in interlocks.
To use it as an interlock, the user should set it in the transition condition G1.
- (2) WAIT must be set to proceed to the next step on operation completion.
However, when there are specifically no conditions to be set as interlocks, set "NOP (No Operation)" in the transition program (Gn).

4.9.2 Selective branch, selective coupling

(1) Selective branch

Executes only the route which condition was judged to have enabled first among the conditions of multiple transitions connected in parallel. Transitions must be all Shifts or WAITs.

(Example) WAIT

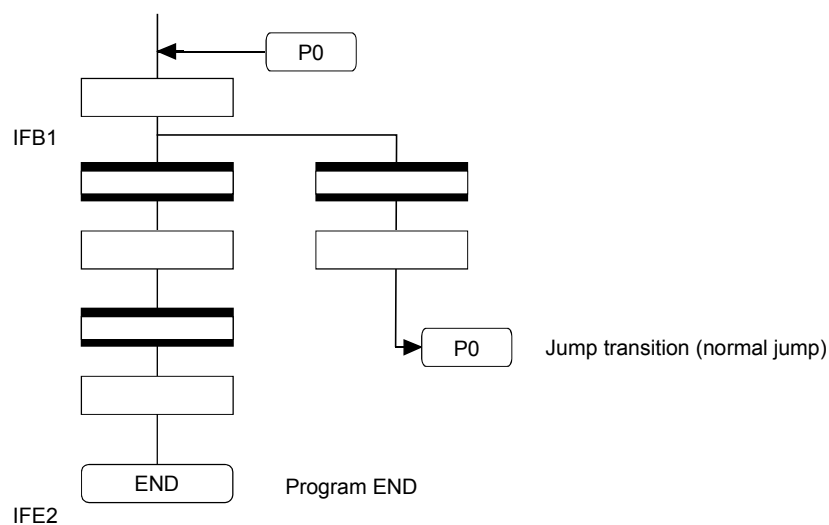


POINT

- (1) Transition condition judgment is not always executed from left to right.
- (2) Using Shift and WAIT together will cause a parallel branch.

(2) Selective coupling

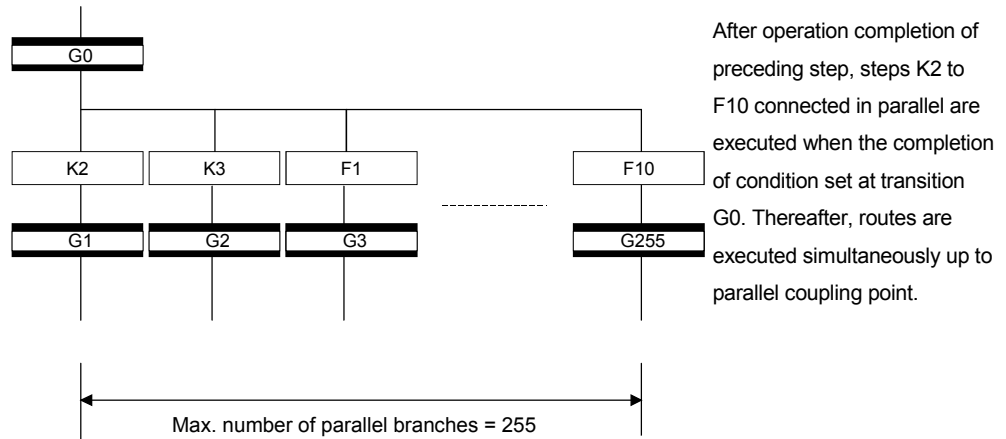
Recoupling of routes into a single route after their processing completions following a selective branch will be a selective coupling. However, you can also make a setting where no coupling will be made as shown below.



4.9.3 Parallel branch, parallel coupling

(1) Parallel branch

Multiple routes connected in parallel are executed simultaneously. Each parallel branch destination may be started by either a step or a transition.



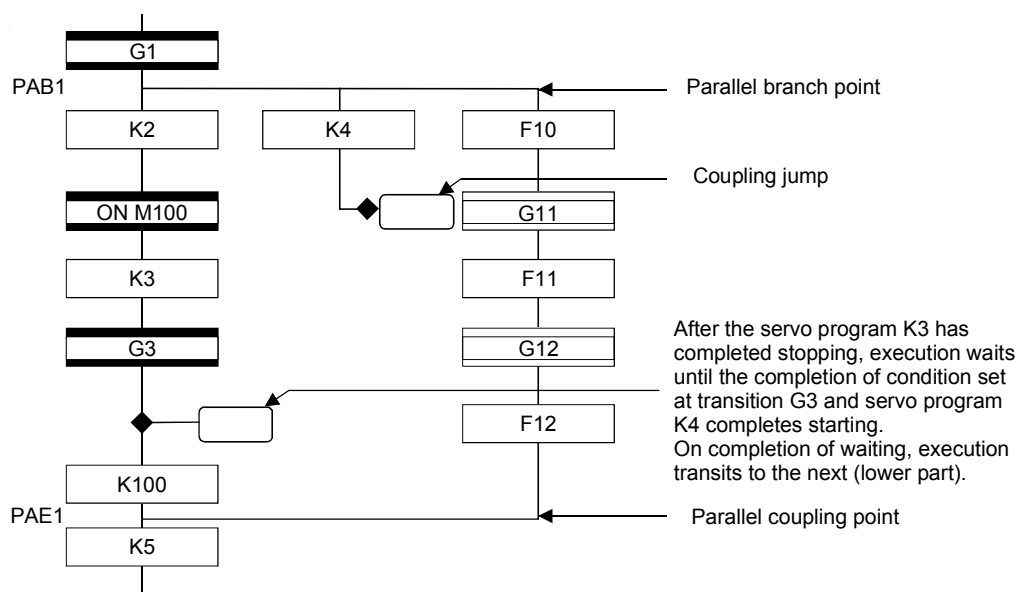
POINT

"Shift" or "WAIT" can be set to a transition preceding a parallel branch.
 "WAITON" and "WAITOFF" cannot be set.

(2) Parallel coupling

A parallel branch must be coupled by a parallel coupling. A jump setting to another branch route can be made within parallel branch-parallel coupling. In this case, a jump destination is a midway parallel coupling point (coupling jump).

You cannot set a jump to exit from within parallel branch-parallel coupling.



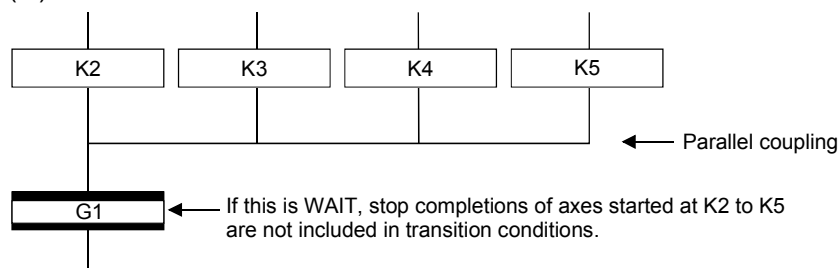
POINT

The number of parallel branches need not match that of couplings at a parallel coupling point.

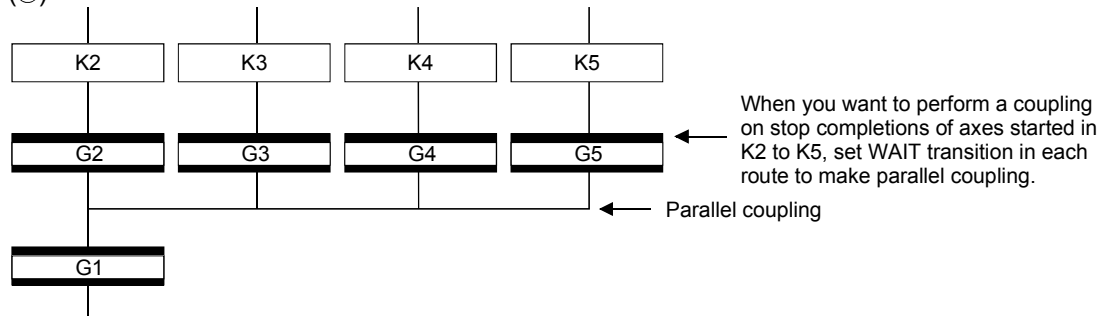
(In the example of the diagram in Section 4.9.3 (2), the number of parallel branches is 3 and that of couplings is 2.)

When a WAIT transition is set right after a parallel coupling, the stop completions of the axes are not included in the waiting conditions if the parallel coupling is preceded by motion control steps. To perform a parallel coupling on stop completions, set WAIT transitions before a parallel coupling.

(×)



(○)



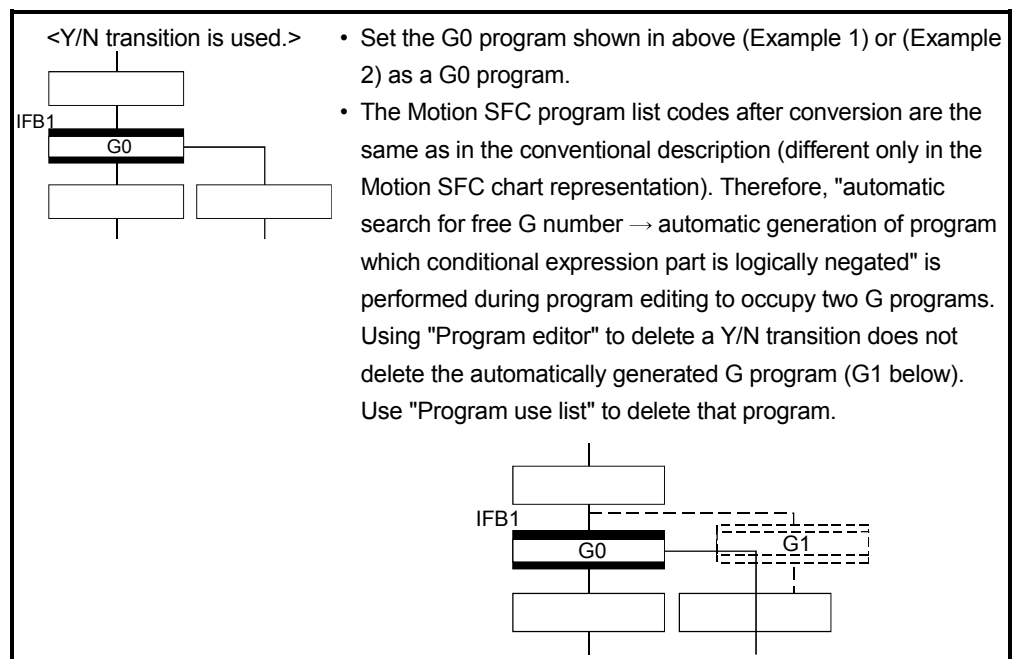
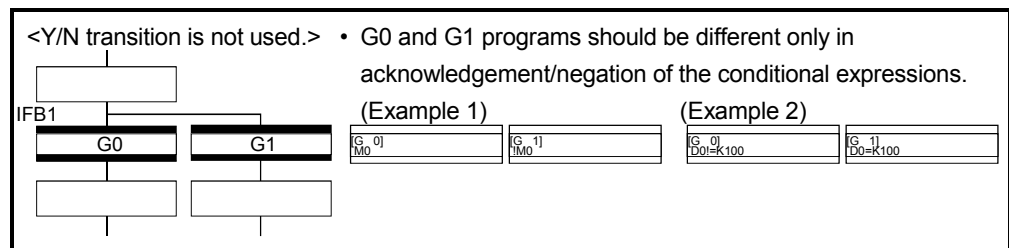
4 MOTION SFC PROGRAMS

4.10 Y/N Transitions

When routes are branch at a transition condition enables and disable, "Shift Y/N transition" or "WAIT Y/N transition" will be useful.

Name	Symbol	Function
Shift Y/N transition		<ul style="list-style-type: none"> When a transition condition set at Gn enables, execution shifts to the lower step. When that condition disables, execution shifts to the right-connected step.
WAIT Y/N transition		<ul style="list-style-type: none"> Differences between "Shift Y/N" and "WAIT Y/N" are the same as those between "Shift" and "WAIT".

A Y/N transition is designed to describe the following two-route selective branch program easily.



(1) Automatic free G number search feature

(a) When not set to automatic numbering

Searches for a free number forward, starting with the "set G number + 1" at the "Shift Y/N" or "WAIT Y/N" symbol.

When no free numbers are found after a search up to 4095, a search is made from 0 to the "set G number - 1".

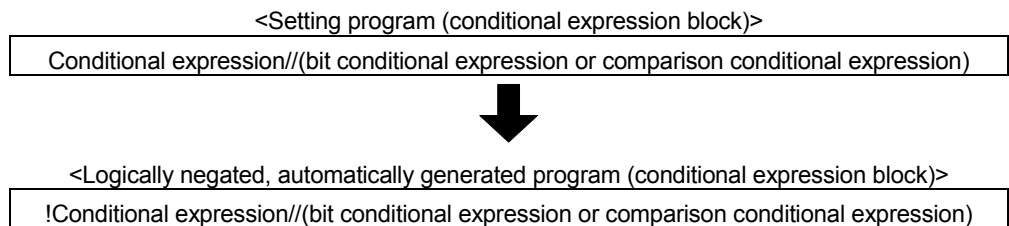
(b) When set to automatic numbering

Searches for a free number forward (or backward) in the automatic numbering range, starting with the "automatically numbered G number + 1 (or -1)" at the "Shift Y/N" or "WAIT Y/N" symbol. (The searching method is as in the automatic numbering setting.)

(2) Automatic logical NOT program generation feature

Automatically generates a program which logically negates the conditional expression block (last block) of the transition program set at "Shift Y/N" or "WAIT Y/N".

The basic is shown below.



Examples are shown below.

<Setting program (conditional expression block)>

(Example 1)

M0 //Bit device ON

(Example 2)

D0!=K100 //Data register D0 is not K100

<Logically negated, automatically generated program (conditional expression block)>

(Example 1)

!(M0) //Bit device OFF

(Example 2)

!(D0!=K100) //Data register D0 is K100

POINT

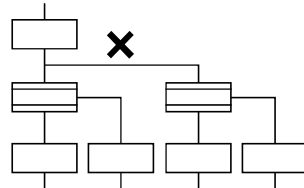
Refer to Section "1.2.3 (2) Table of the operation control/transition instruction" for the instructions usable in the conditional expressions of "Shift Y/N" or "WAIT Y/N" transition programs.

(3) Instructions for the Motion SFC charts

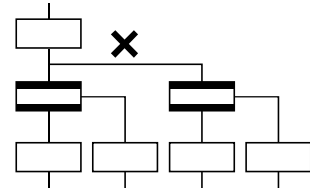
Any Motion SFC chart that will be meaningless to or conflict with the definition of Y/N transitions will result in an error at the time of editing (or Motion SFC chart conversion). Their patterns and instructions will be given below.

(a) When "Shift Y/N" or "WAIT Y/N" is connected as a selective branch or parallel branch: Error

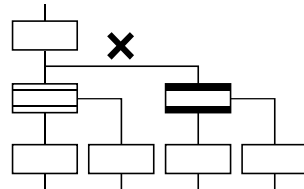
- "Shift Y/N" used as selective branch



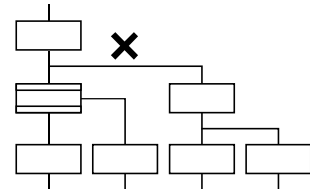
- "WAIT Y/N" used as selective branch



- "Shift Y/N" and "WAIT Y/N" used as parallel branch

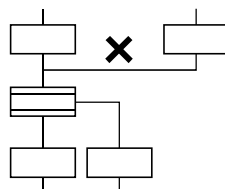


- "Shift (or WAIT) Y/N" used with other step/transition as parallel branch or selective branch

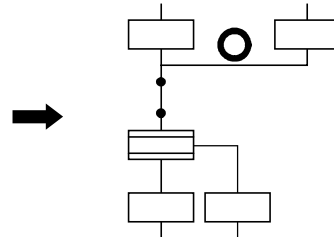


(b) When a coupling precedes "Shift Y/N" or "WAIT Y/N": Provide "coupling-branch continuation" in between.

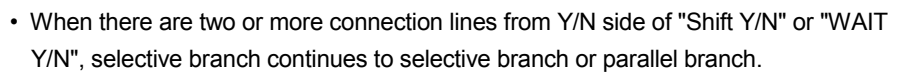
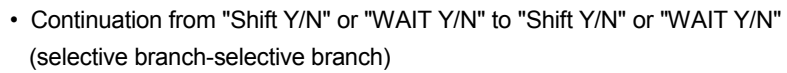
- Direct coupling with "Shift Y/N" or "WAIT Y/N" is not allowed.



- Provide "coupling-branch continuation" in between.



- End (END) from "Shift Y/N" or "WAIT Y/N"
- Jump from "Shift Y/N" or "WAIT Y/N"



4 MOTION SFC PROGRAMS

4.11 Motion SFC Comments

A comment can be set to each symbol of the step/transition in the motion SFC chart. Comments are shown in the Motion SFC chart by changing the display mode to "Comment display" on the Motion SFC program edit screen. Since the Motion SFC comments are stored into the CPU code area, performing read from PC displays the Motion SFC chart with comments.

Classification	Name	Symbol	Comment Setting
Program start/end	START		Comment setting cannot be made.
	END		
Step	Motion control step		
	Once execution type operation control step		
	Scan execution type operation control step		
	Subroutine call/start step		
	Clear step		
Transition	Shift (preread transition)		Up to 80 characters Displayed in 20 characters × 4 lines
	WAIT		
	WAITON		
	WAITOFF		
	Shift Y/N		
	WAIT Y/N		
Jump	Jump		Up to 64 characters Displayed in 16 characters × 4 lines
Pointer	Pointer		

POINT	
(1)	Motion SFC comments are stored into the CPU code area. The CPU code area stores the Motion SFC chart codes, operation control (F/FS) program codes, transition (G) program codes and Motion SFC comments. Be careful not to set too many comments to avoid code area overflow. (Refer to Section "1.2.2 (1) (b) Motion SFC Performance Specifications" for the code area sizes.)
(2)	You cannot use "," in comment statements.

5. OPERATION CONTROL PROGRAMS

Refer to Section "12.3 Motion SFC Error Code List" for error codes of the operation error.

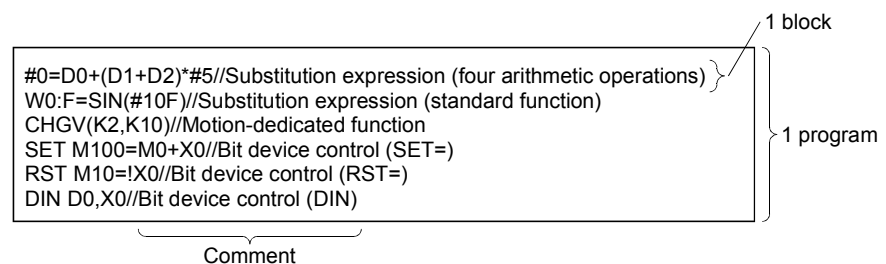
(Refer to the "Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" and "Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for minor errors of the operation error.)

5.1 Operation Control Programs

(1) Operation control programs

- (a) Substitution operation expressions, motion-dedicated functions and bit device control commands can be set in operation control program.
- (b) Multiple blocks in one operation control program can be set.
- (c) There are no restrictions on the number of blocks that may be set in one operation control program. However, one program is within 64k bytes.
- (d) The maximum number of characters in one block is 128.
- (e) Transition conditions cannot be set. Transition conditions can be set only in transition programs.
- (f) The bit conditional expression that logical data value (true or false) is returned in an operation control program, a comparison conditional expression can be set up only as a source (S) of device set (SET=) or device reset (RST=).

An operation control program example is shown below.



(2) Priorities of operators and functions

Operators and functions have the following priorities.

Using parentheses allows an operation sequence to be specified freely.

Priority	Item (Operator, Function)
<div>High</div> <div>↑</div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div>↓</div> <div>Low</div>	Calculation within parentheses ((...))
	Standard function (SIN, COS, etc.),
	Type conversion (USHORT, LONG, etc.)
	Bit inversion (~), logical negation (!), sign inversion (—)
	Multiplication (*), division (/), remainder (%)
	Addition (+), subtraction (—)
	Bit left shift (<<), bit right shift (>>)
	Comparison operators: Less than (<), less than or equal to (<=), more than (>), more than or equal to (>=)
	Comparison operators: Equal to (==), not equal to (!=)
	Bit logical AND (&)
	Bit exclusive OR (^)
	Bit logical OR ()
	Logical AND (*)
	Logical OR (+)
	Substitution (=)

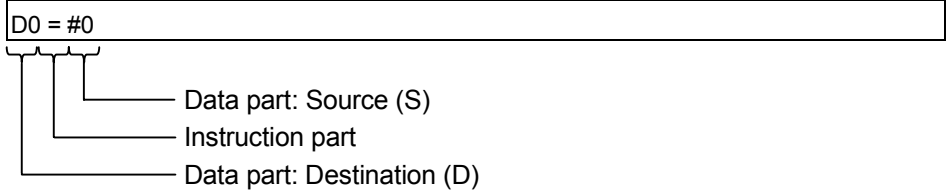
(3) Structure of instruction

Many of the instructions usable in operation control programs can be divided into instruction and data parts.

The instruction and data parts are used for the following purposes.

- Instruction part..... Indicates the function of that instruction.
- Data part..... Indicates the data used in the instruction.

"Substitution: =" structure example



(a) Source (S)

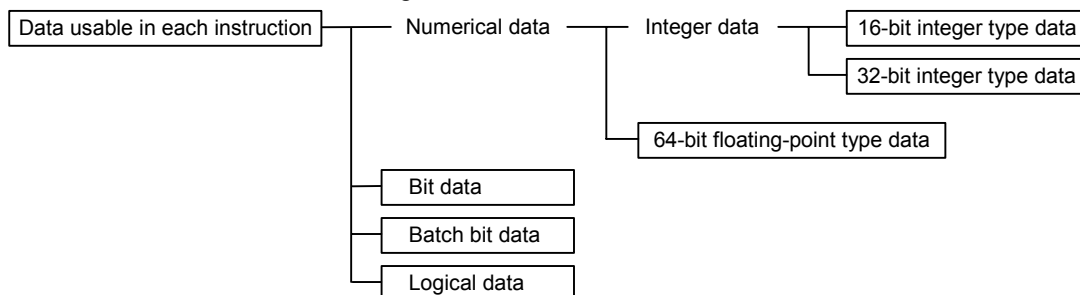
- 1) The source is the data used in an operation.
- 2) It varies with the device specified in each instruction is shown below.
 - Bit or word device
Specify the device which stores the data used in operation.
The data must have been stored in the specified device until the operation is executed.
Changing the data stored in the specified device during program execution allows changing the data used in that instruction.
 - Constant
Specify the numerical value used in an operation.
As the constant is set during program creation, it cannot be changed during program running.

(b) Destination (D)

- 1) As the destination data, after-operation data is stored.
- 2) Destination data is always set the device for storing the data.

(4) How to specify data

There are the following six different data usable in each instruction.



(a) 16-bit integer type data

The 16-bit integer type data is 16-bit integer value data.

Word devices are used in increments of 1 point.

Data ranges are shown below.

	Decimal representation	Hexadecimal representation
Data range	K-32768 to K32767	H0000 to HFFFF

(b) 32-bit integer type data

The 32-bit integer type data is 32-bit integer value data.

Word devices are used in increments of 2 points: (specified device No.), (specified device No.+1). Data ranges are shown below.

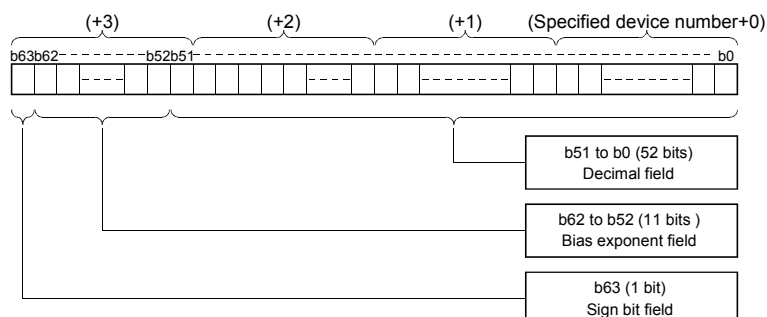
	Decimal representation	Hexadecimal representation
Data range	K-2147483648L to K2147483647L	H00000000L to HFFFFFFFL

(c) 64-bit floating-point type data

The 64-bit floating-point type data is IEEE-formatted, 64-bit floating-point value data.

Word devices are used in increments of 4 points: (specified device No.), (specified device No.+1), (specified device No.+2), (specified device No.+3).

1) The internal bit locations are shown below.



2) The represented value is shown below. (The bias value is H3FF.)

$$(-1)^{[\text{Sign bit field}]} * (1.0 + [\text{decimal field}]) * 2^{([\text{Bias exponent field}] - [\text{bias value}])}$$

5 OPERATION CONTROL PROGRAMS

3) Data ranges are shown below.

	Decimal representation	Hexadecimal representation
Data range	K-1.79E+308 to K-2.23E-308, K0.0, K2.23E-308 to K1.79E+308	H0000000000000000, H0010000000000000 to H7FE1CCF385EBC89F, H8000000000000000, H8010000000000000 to HFFE1CCF385EBC89F

4) A round-off error may be produced in a 64-bit floating-point type data operation. Especially when using 64-bit floating-point type data in a comparison operation, note that a round-off error may cause an intended operation.

Example) In the following transition program, the result of the comparison operation may not become true depending on the value of #200F due to a round-off error.

```
#100F=SQRT(#200F)
#300F=#100F*#100F
#200F==#300F
```

(d) Bit data

The bit data is the data where a contact/coil or similar device is handled in increments of 1 bit. It is used in device set (SET=) and device reset (RST=).

Example 1

```
SET M0
```

└── Bit data

(e) Batch bit data

The batch bit data is the data where bit data is handled in increments of 16/32 points. It is used in device input (DIN) and device output (DOUT). As indicated below, whether the bit data is handled in increments of 16 or 32 points is governed by the data type of the word device used as an input destination/output source.

	Increments of 16 points	Increments of 32 points
Program example	DIN #0, M0 DOUT M0, D0	DIN #0L, M0 DOUT M0, D0L
Used devices	(Specified device No.) to (specified device No.+15) M0 to M15 in the above program example	(Specified device No.) to (specified device No.+31) M0 to M31 in the above program example

5 OPERATION CONTROL PROGRAMS

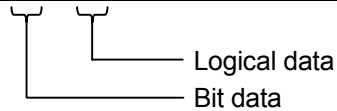
(f) Logical data

The logical data is a value returned by a bit or comparison conditional expression and indicates whether the result is true or false.

Normally, it is used in the conditional expression of a transition program. In an operation control program, the logical data is used in a bit conditional expression set to device set (SET=) or device reset (RST=).

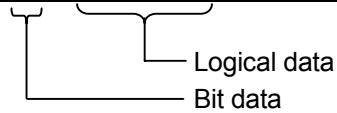
Example 1

SET M0 = X10



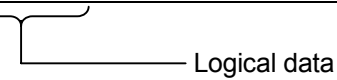
Example 2

RST M5 = !X10 * M100



Example 3 (transition program)

D0 == K100



5 OPERATION CONTROL PROGRAMS

5.2 Device Descriptions

Word and bit device descriptions are shown below.

(1) Word device descriptions

	Device descriptions			Device No. (n) specified ranges
	16-bit integer type	32-bit integer type ("n" is even No.)	64-bit floating-point type ("n" is even No.)	
Data register	Dn	DnL	DnF	0 to 8191
Link register	Wn	WnL	Wn:F	0 to 1FFF
Special register	SDn	SDnL	SDnF	0 to 2255
Motion device	#n	#nL	#nF	0 to 8735 (Motion SFC dedicated devices: 8000 to 8735)
Multiple CPU area device	U□\Gn	U□\GnL	U□\GnF	10000 to (10000+p-1) ^(Note-1) □: CPU No. (No.1: 3E0, No.2: 3E1, No.3: 3E2, No.4: 3E3) CPU No. that is larger than the number of Multiple CPU cannot be set.
Coasting timer	—	FT	—	—

(Note-1): "p" indicates the user setting area points of the Multiple CPU high speed transmission area for the each CPU.

- (a) For differentiation, the 32-bit floating-point type is ended by L and the 64-bit floating-point type by F (F for the link register).
- (b) For the 32-bit integer type and 64-bit floating-point type, specify the device number with an even number. (It cannot be set as an odd number).
- (c) The coasting timer FT is incremented per 888[μs]. (The coasting timer is a 32-bit integer type.)

(2) Bit device descriptions

	Device description	Device No. (n) specified ranges
Input relay	Xn/PXn	0 to 1FFF
Output relay	Yn/PYn	0 to 1FFF
Internal relay	Mn	0 to 8191
Multiple CPU area device	U□\Gn	10000.0 to (10000+p-1).F ^(Note-1) □: CPU No. (No.1: 3E0, No.2: 3E1, No.3: 3E2, No.4: 3E3) CPU No. that is larger than the number of Multiple CPU cannot be set.
Link relay	Bn	0 to 1FFF
Annunciator	Fn	0 to 2047
Special relay	SMn	0 to 2255

(Note-1): "p" indicates the user setting area points of the Multiple CPU high speed transmission area for the each CPU.

- (a) When using the device in DIN or DOUT as batch bit data, specify "n" as a multiple of 16.
- (b) When using the device in Multiple CPU area device as batch bit data, specify it as word device without making bit specification.

(3) Indirect specification of device No.

In the above word/bit device descriptions, device No. (n) can be specified indirectly.

- (a) Indirect specification of device No. (n) using word device
 - The word device which the device No. was specified indirectly cannot be used.
 - You can use the 16-bit and 32-bit integer type word devices for indirect specification.
The 64-bit floating-point type cannot be used.

(Description examples)

Good example	Bad example
#(D10)	#(D(D5))
D(#10L)F	D(#4F)

- (b) Indirect specification of device No. (n) using word device using operation expression
 - Device No. can be specified indirectly by calculation expressions which use the following data and operators.

Usable data	16-bit integer type word device
	32-bit integer type word device
	16-bit integer type constant
	32-bit integer type constant
Usable operators	Addition: +
	Subtraction: —
	Multiplication: *
	Division: /
	Remainder: %
	Sign inversion: —

- The word device which the device No. is specified indirectly cannot be used.
- Only one operator may be used.

5 OPERATION CONTROL PROGRAMS

(Description examples)

Good example	Bad example
#(D10-K5)	#(D(D5)F+K20)
D(#10L%H6L)F	D(#4L<<K2)

(Note) : When you want to use the result of calculation other than the above to specify the device No. indirectly, describe it in two blocks as shown below.

D0=SHORT(ASIN(#0F)) W0=#(D0)

POINT

Refer to Chapter 2 of the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for the user setting area points of the Multiple CPU high speed transmission area.
--

5 OPERATION CONTROL PROGRAMS

5.3 Constant Descriptions

The constant descriptions of the 16-bit integer type, 32-bit integer type and 64-bit floating-point type are shown below.

	16-Bit integer type	32-Bit integer type	64-Bit floating-point type
Decimal representation	K-32768 to K32767	K-2147483648L to K2147483647L	K-1.79E+308 to K-2.23E-308, K0.0, K2.23E-308 to K1.79E+308
Hexadecimal representation	H0000 to HFFFF	H00000000L to HFFFFFFFFL	—

- (1) The 32-bit integer type is ended by L and the 64-bit floating-point type is provided with a decimal point and exponent part (E) to denote their data types explicitly.
- (2) The constant without the data type is regarded as the applicable minimum type.
- (3) The constant in decimal representation is headed by K and the one in hexadecimal representation by H.
K can be omitted.
- (4) The 64-bit floating-point type cannot be represented in hexadecimal.

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.4 Binary Operations

5.4.1 Substitution : =

Format	(D)=(S)	Number of basic steps	4
--------	---------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(D)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—	—	—	—	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Word device/constant/calculation expression to be substituted	Data type of (D)
(D)	Word device which will store the operation result	

[Functions]

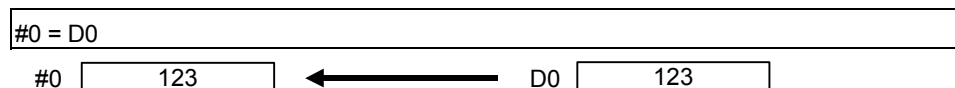
- The data value specified with (S) is substituted to the specified word device at (D).
- When (S) and (D) differ in data type, the data at (S) is converted into the data type of (D) and the resultant data is substituted.
(When (D) is a 16- or 32-bit integer type and (S) is a 64-bit floating-point type, the fraction part of (S) is discarded.)

[Errors]

- An operation error will occur if:
 - The data at (S) is outside the data type range of (D); or
 - (D) or (S) is an indirectly specified device and its device No. is outside the range.

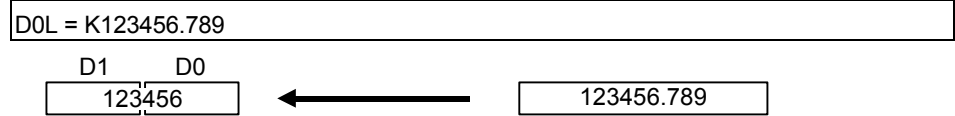
[Program examples]

- Program which substitutes the D0 value to #0



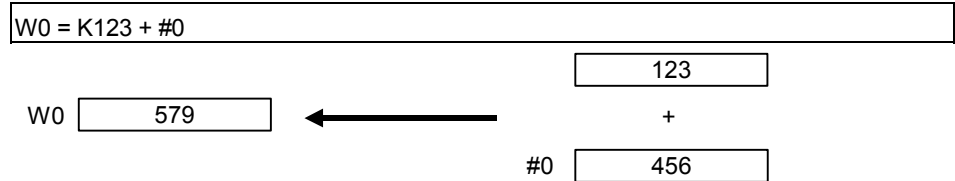
5 OPERATION CONTROL PROGRAMS

- (2) Program which substitutes K123456.789 to D0L



The 64-bit floating-point type is converted into the 32-bit integer type and the result is substituted.

- (3) Program which substitutes the result of adding K123 and #0 to W0



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.4.2 Addition : +

Format	(S1)+(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Augend data	Data type of (S1) or (S2) which is greater
(S2)	Addend data	

[Functions]

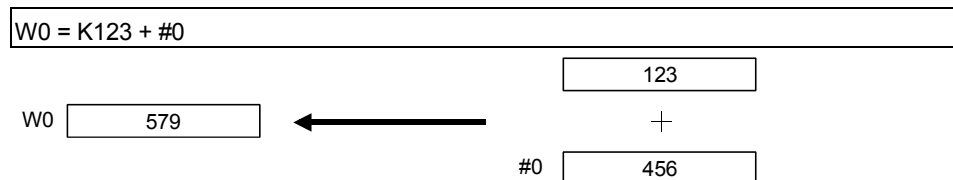
- (1) The data specified with (S2) is added to the data specified with (S1).
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

[Errors]

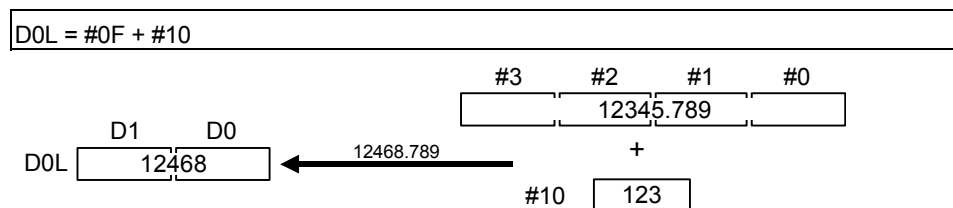
- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which substitutes the result of adding K123 and #0 to W0



- (2) Program which substitutes the result of adding #0F and #10 to D0L



The 64-bit floating-point type data are used for addition, and the result is converted into the 32-bit integer type and then substituted.

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.4.3 Subtraction : —

Format	(S1) — (S2)	Number of basic steps	4
--------	-------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Minuend data	Data type of (S1) or (S2) which is greater
(S2)	Subtracted data	

[Functions]

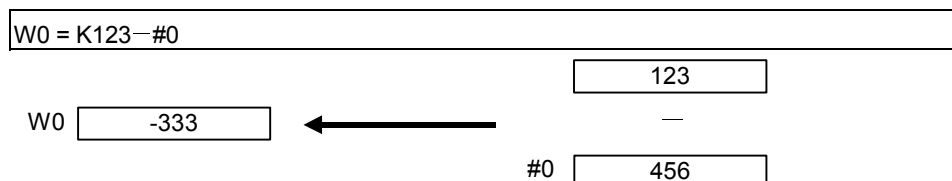
- (1) The data specified with (S2) is subtracted from the data specified with (S1).
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

[Errors]

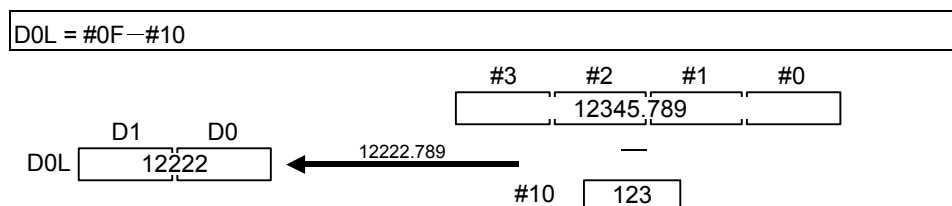
- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which substitutes the result of subtracting #0 from K123 to W0



- (2) Program which substitutes the result of subtracting #10 from #0F to D0L



The 64-bit floating-point type data are used for subtraction, and the result is converted into the 32-bit integer type and then substituted.

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.4.4 Multiplication : *

Format	(S1)*(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Multiplicand data	Data type of (S1) or (S2) which is greater
(S2)	Multiplier data	

[Functions]

- (1) The data specified with (S1) is multiplied by the data specified with (S2).
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

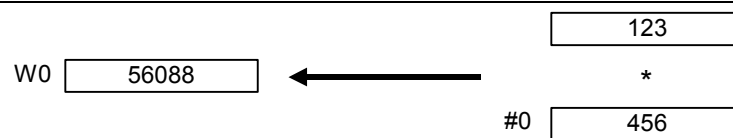
[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

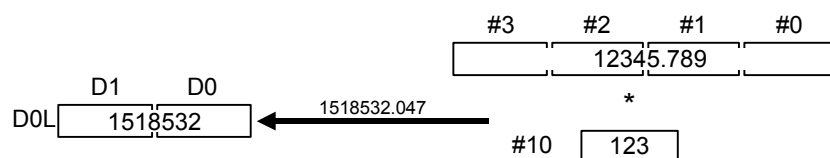
- (1) Program which substitutes the result of multiplying K123 by #0 to W0

W0 = K123 * #0



- (2) Program which substitutes the result of multiplying #0F by #10 to D0L

D0L = #0F * #10



The 64-bit floating-point type data are used for multiplication, and the result is converted into the 32-bit integer type and then substituted.

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.4.5 Division : /

Format	(S1)/(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Dividend data	Data type of (S1) or (S2) which is greater
(S2)	Divisor data	

[Functions]

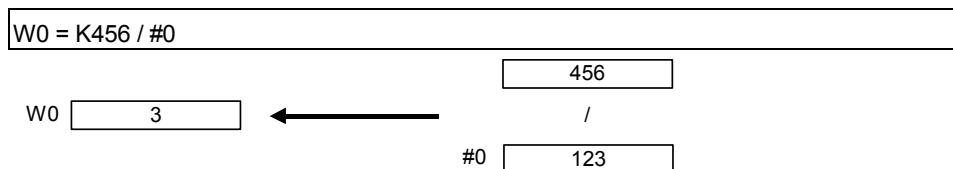
- (1) The data specified with (S1) is divided by the data specified with (S2) to find a quotient.
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

[Errors]

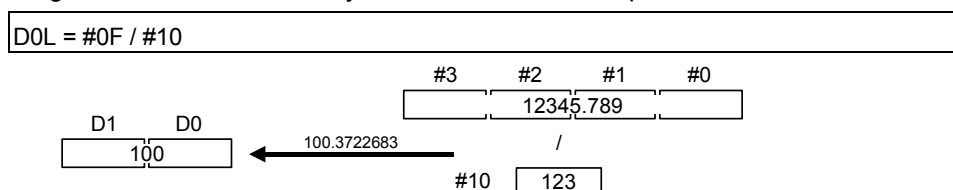
- (1) An operation error will occur if:
 - (S2) is 0; or
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which divides K456 by #0 and substitutes a quotient to W0



- (2) Program which divides #0F by #10 and substitutes a quotient to D0L



The 64-bit floating-point type data are used for division, and the quotient is converted into the 32-bit integer type and then substituted.

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.4.6 Remainder : %

Format	(S1)%(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Dividend data	Data type (integer type) of (S1) or (S2) which is greater (Integer type)
(S2)	Divisor data	

[Functions]

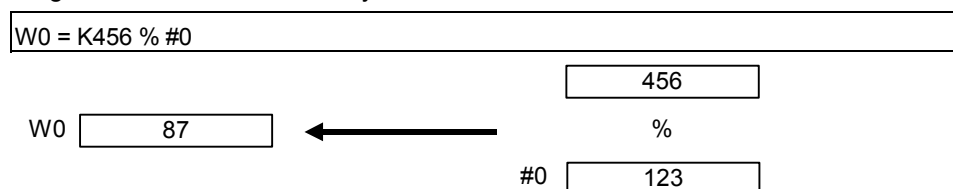
- (1) The data specified with (S1) is divided by the data specified with (S2) to find a remainder.
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S2) is 0; or
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which divides K456 by #0 and substitutes a remainder to W0



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.5 Bit Operations

5.5.1 Bit inversion (Complement) : ~

Format	~ (S)	Number of basic steps	2
--------	-------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose bits will be inverted	Data type of (S) (Integer type)

[Functions]

- (1) The bit inverted value of the data specified with (S) is found.

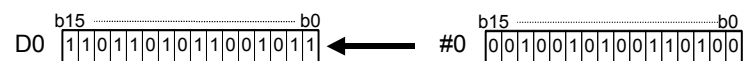
[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which finds the bit inverted value of #0 and substitutes the value to D0

D0 = ~ #0



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.5.2 Bit logical AND : &

Format	(S1)&(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be ANDed bit-by-bit	Data type of (S1) or (S2) which is greater (Integer type)
(S2)		

[Functions]

- (1) The bit-by-bit logical product of the data specified with (S1) and the data specified with (S2) is found.
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed. At this time, note that signed data is converted.

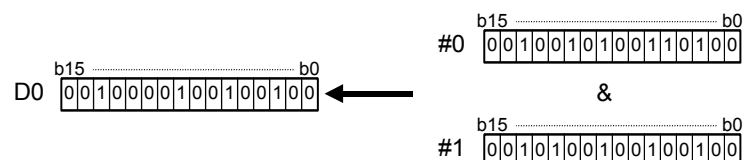
[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which ANDs #0 and #1 and substitutes the result to D0

D0 = #0 & #1



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.5.3 Bit logical OR : |

Format	(S1) (S2)	Number of basic steps	4
--------	-------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be ORed bit-by-bit	Data type of (S1) or (S2) which is greater (Integer type)
(S2)		

[Functions]

- (1) The bit-by-bit logical add of the data specified with (S1) and the data specified with (S2) is found.
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed. At this time, note that signed data is converted.

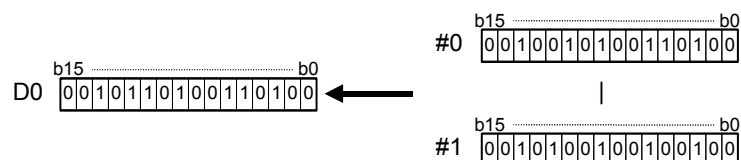
[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which ORs #0 and #1 and substitutes the result to D0

D0 = #0 | #1



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.5.4 Bit exclusive logical OR : ^

Format	(S1)^(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be EXCLUSIVE ORed bit-by-bit	Data type of (S1) or (S2) which is greater (Integer type)
(S2)		

[Functions]

- (1) The bit-by-bit exclusive logical add of the data specified with (S1) and the data specified with (S2) is found.
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed. At this time, note that signed data is converted.

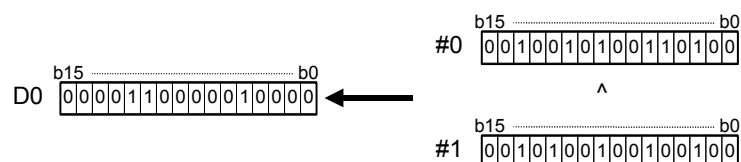
[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which EXCLUSIVE ORs #0 and #1 and substitutes the result to D0

D0 = #0 ^ #1



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.5.5 Bit right shift : >>

Format	(S1) >> (S2)	Number of basic steps	4
--------	--------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data to be right-shifted	Data type of (S1) (Integer type)
(S2)	Number of right shifts	

[Functions]

- (1) The data specified with (S1) is shifted to the right by the number of times specified with (S2).
- (2) If the most significant bit of (S1) is 1, 1 enters the most significant bit of the right shift result.
If the most significant bit of (S1) is 0, 0 enters the most significant bit of the right shift result.
- (3) When (S1) is a 16-bit integer type and (S2) is a negative number or not less than 16, the result is 0.
- (4) When (S1) is a 32-bit integer type and (S2) is a negative number or not less than 32, the result is 0.

[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which shifts #0 two bit positions to the right and substitutes the result to D0

D0 = #0 >> K2



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.5.6 Bit left shift : <<

Format	(S1) << (S2)	Number of basic steps	4
--------	--------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data to be left-shifted	Data type of (S1) (Integer type)
(S2)	Number of left shifts	

[Functions]

- (1) The data specified with (S1) is shifted to the left by the number of times specified with (S2).
- (2) 0 enters the least significant bit of the left shift result.
- (3) When (S1) is a 16-bit integer type and (S2) is a negative number or not less than 16, the result is 0.
- (4) When (S1) is a 32-bit integer type and (S2) is a negative number or not less than 32, the result is 0.

[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which shifts #0 one bit position to the left and substitutes the result to D0

D0 = #0 << K1	
D0	#0
b15 b0 0 1 0 0 1 0 1 0 1 0 0 1 1 0 1 0 0 0	b15 b0 0 0 1 0 0 1 0 1 0 1 0 0 1 1 0 1 0 0

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.5.7 Sign inversion (Complement of 2) : —

Format	—(S)
--------	------

Number of basic steps	2
-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose sign will be inverted	Data type of (S)

[Functions]

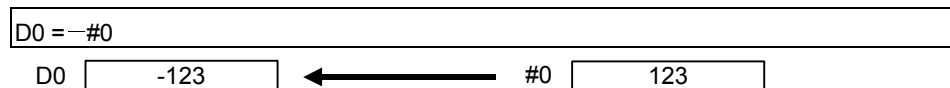
- (1) The sign-inverted value of the data specified with (S) is found.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which substitutes the sign-inverted value of #0 to D0



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6 Standard Functions

5.6.1 Sine : SIN

Format	SIN(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Angle data on which SIN (sine) operation will be performed	Floating-point type

[Functions]

- (1) SIN (sine) operation is performed on the data specified with (S).
- (2) The data specified with (S) is in an angle (degree) unit.
- (3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which performs the SIN operation of D0 and substitutes the result to #0F

#0F = SIN(D0)				
#3	#2	#1	#0	
0.70710678118655				← D0 45

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.2 Cosine : COS

Format	COS(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Angle data on which COS (cosine) operation will be performed	Floating-point type

[Functions]

- (1) COS (cosine) operation is performed on the data specified with (S).
- (2) The data specified with (S) is in an angle (degree) unit.
- (3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which performs the COS operation of D0 and substitutes the result to #0F

#0F = COS(D0)



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.3 Tangent : TAN

Format	TAN(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Angle data on which TAN (tangent) operation will be performed	Floating-point type

[Functions]

- (1) TAN (tangent) operation is performed on the data specified with (S).
- (2) The data specified with (S) is in an angle (degree) unit.
- (3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range; or
 - (S) is $90 + (180 \times n)$. ("n" is an integer)

[Program examples]

- (1) Program which performs the TAN operation of D0 and substitutes the result to #0F

```
#0F = TAN(D0)
```



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.4 Arcsine : ASIN

Format	ASIN(S)	Number of basic steps	2
--------	---------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	SIN value data on which SIN^{-1} (arcsine) operation will be performed	Floating-point type

[Functions]

- (1) SIN^{-1} (arcsine) operation is performed on the SIN value data specified with (S) to find an angle.
- (2) The SIN value specified with (S) must be within the range -1.0 to 1.0.
- (3) The operation result is in an angle (degree) unit.
- (4) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is outside the range -1.0 to 1.0; or
 - (S) is an indirectly specified device and its device number is outside the range.

[Program examples]

- (1) Program which performs the SIN^{-1} (arcsine) operation of D0 and substitutes the result to #0F

#0F = ASIN(D0)				
#3	#2	#1	#0	
		90.0		
				← D0
				1

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.5 Arccosine : ACOS

Format	ACOS(S)	Number of basic steps	2
--------	---------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	COS value data on which COS^{-1} (arccosine) operation will be performed	Floating-point type

[Functions]

- (1) COS^{-1} (arccosine) operation is performed on the COS value data specified with (S) to find an angle.
- (2) The COS value specified with (S) must be within the range -1.0 to 1.0.
- (3) The operation result is in an angle (degree) unit.
- (4) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is outside the range -1.0 to 1.0; or
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which performs the COS^{-1} (arccosine) operation of D0F and substitutes the result to #0F

#0F = ACOS(D0F)											
#3	#2	#1	#0			D3	D2	D1	D0		
			60.0						0.5		



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.6 Arctangent : ATAN

Format	ATAN(S)	Number of basic steps	2
--------	---------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	TAN value data on which TAN^{-1} (arctangent) operation will be performed	Floating-point type

[Functions]

- (1) TAN^{-1} (arctangent) operation is performed on the TAN value data specified with (S) to find an angle.
- (2) The operation result is in an angle (degree) unit.
- (3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which performs the TAN^{-1} (arctangent) operation of D0F and substitutes the result to #0F

#0F = ATAN(D0F)											
#3	#2	#1	#0		D3	D2	D1	D0			
			45.0					1.0			

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.7 Square root : SQRT

Format	SQRT(S)	Number of basic steps	2
--------	---------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data on which square root operation will be performed	Floating-point type

[Functions]

- (1) The square root of the data specified with (S) is found.
- (2) Only a positive number may be specified with (S). (Operation cannot be performed with a negative number.)
- (3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is a negative number; or
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which finds the square root of D0F and substitutes the result to #0F

#0F = SQRT(D0F)											
#3	#2	#1	#0		D3	D2	D1	D0			
		3.0					9.0				

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.8 Natural logarithm : LN

Format	LN(S)	Number of basic steps	2
--------	-------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data on which natural logarithm operation will be performed	Floating-point type

[Functions]

- (1) The base e natural logarithm of the data specified with (S) is found.
- (2) Only a positive number may be specified with (S). (Operation cannot be performed with a negative number.)
- (3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is 0 or a negative number; or
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which finds the natural logarithm of D0F and substitutes the result to #0F

#0F = LN(D0F)

#3	#2	#1	#0		D3	D2	D1	D0
	2	302585	0929940	←			10.0	

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.9 Exponential operation : EXP

Format	EXP(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data on which exponential operation will be performed	Floating-point type

[Functions]

- (1) Exponential operation is performed on the base e data specified with (S).
- (2) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which performs exponential operation of D0F and substitutes the result to #0F

#0F = EXP(D0F)											
#3	#2	#1	#0					D3	D2	D1	D0
	4	4	2	1	3	.	3	9	2	0	0
									1	3	.

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.10 Absolute value : ABS

Format	ABS(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data on which absolute value conversion will be performed	Data type of (S)

[Functions]

- (1) The absolute value of the data specified with (S) is found.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which finds the absolute value of D0F and substitutes the result to #0F

#0F = ABS(D0F)											
#3	#2	#1	#0		D3	D2	D1	D0			
			33.0					-33.0			

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.11 Round-off : RND

Format	RND(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose fractional portion will be rounded off	Data type of (S)

[Functions]

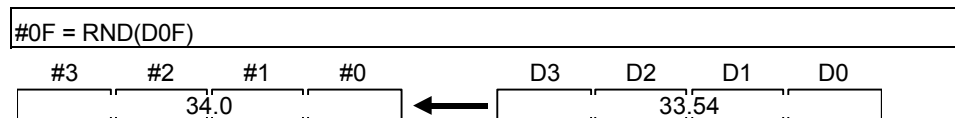
- (1) The rounded-off fractional portion value of the data specified with (S) is found.
- (2) If (S) is a negative number, the absolute value of (S) is found and its fractional portion is rounded off and signed.
- (3) If (S) is an integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

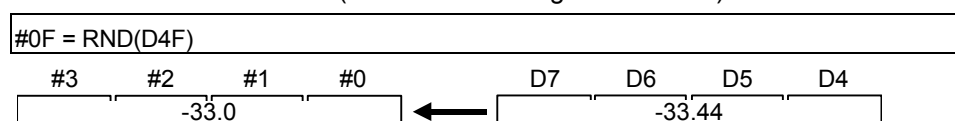
- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which finds the rounded-off fractional portion value of D0F and substitutes the result to #0F



- (2) Program which finds the rounded-off fractional portion value of D4F and substitutes the result to #0F (when D4F is a negative number)



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.12 Round-down : FIX

Format	FIX(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose fractional portion will be rounded down	Data type of (S)

[Functions]

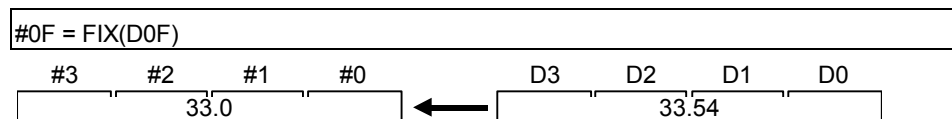
- (1) The largest integer not greater than the data specified with (S) is found.
- (2) If the (S) value is positive, the absolute value will be smaller, and if it is negative, the absolute value will be greater.
- (3) If (S) is an integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

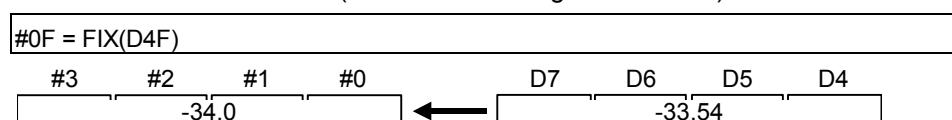
- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which finds the rounded-down fractional portion value of D0F and substitutes the result to #0F



- (2) Program which finds the rounded-down fractional portion value of D4F and substitutes the result to #0F (when D4F is a negative number)



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.13 Round-up : FUP

Format	FUP(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose fractional portion will be rounded up	Data type of (S)

[Functions]

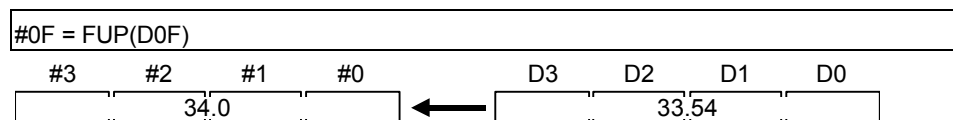
- (1) The smallest integer not less than the data specified with (S) is found.
- (2) If the (S) value is positive, the absolute value will be greater, and if it is negative, the absolute value will be smaller.
- (3) If (S) is an integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

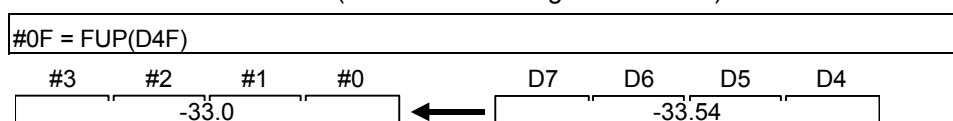
- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which finds the rounded-up fractional portion value of D0F and substitutes the result to #0F



- (2) Program which finds the rounded-up fractional portion value of D4F and substitutes the result to #0F (when D4F is a negative number)



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.14 BCD → BIN conversion : BIN

Format	BIN(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	BCD data which will be converted into BIN data	Data type of (S) (Integer type)

[Functions]

- (1) The BCD data specified with (S) is converted into BIN data.
- (2) If (S) is a 16-bit integer type, the data range is 0 to 9999.
- (3) If (S) is a 32-bit integer type, the data range is 0 to 99999999.

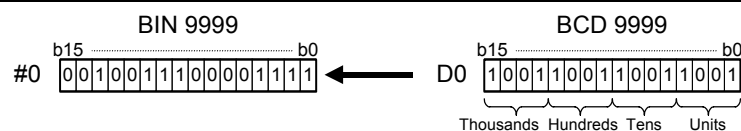
[Errors]

- (1) An operation error will occur if:
 - A value other than 0 to 9 is in any digit of (S); or
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which converts the BCD data of D0 into BIN data and substitutes the result to #0

#0 = BIN(D0)



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.6.15 BIN → BCD conversion : BCD

Format	BCD(S)	Number of basic steps	2
--------	--------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	BIN data which will be converted into BCD data	Data type of (S) (Integer type)

[Functions]

- (1) The BIN data specified with (S) is converted into BCD data.
- (2) If (S) is a 16-bit integer type, the data range is 0 to 9999.
- (3) If (S) is a 32-bit integer type, the data range is 0 to 99999999.

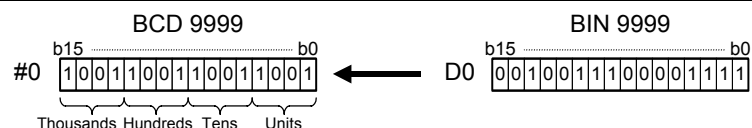
[Errors]

- (1) An operation error will occur if:
 - The data is other than 0 to 9999 when (S) is a 16-bit integer type;
 - The data is other than 0 to 99999999 when (S) is a 32-bit integer type; or
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which converts the BIN data of D0 into BCD data and substitutes the result to #0

#0 = BCD(D0)



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.7 Type Conversions

5.7.1 Signed 16-bit integer value conversion : SHORT

Format	SHORT(S)	Number of basic steps	2
--------	----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into signed 16-bit integer value	16-bit integer type

[Functions]

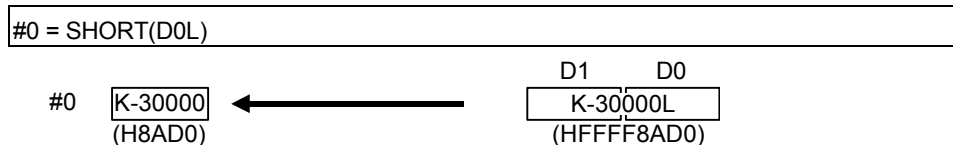
- (1) The data specified with (S) is converted into a signed 16-bit integer value.
- (2) The data range of (S) is -32768 to 32767.
- (3) When (S) is a 64-bit floating-point type, its fractional portion is rounded down before conversion is made.
- (4) If (S) is a 16-bit integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

- (1) An operation error will occur if:
 - The (S) data is outside the range -32768 to 32767; or
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which converts the data of D0L into a signed 16-bit integer value and substitutes the result to #0



5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.7.2 Unsigned 16-bit integer value conversion : USHORT

Format	USHORT(S)	Number of basic steps	2
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into unsigned 16-bit integer value	16-bit integer type

[Functions]

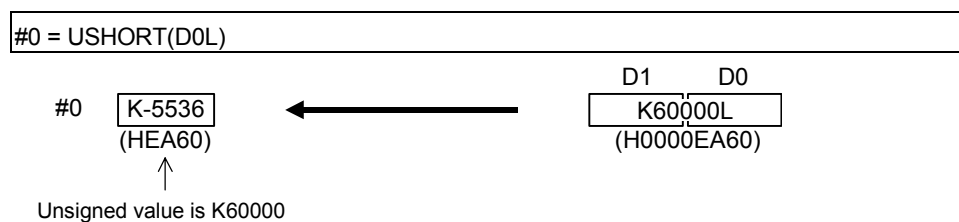
- (1) The data specified with (S) is converted into an unsigned 16-bit integer value.
- (2) The data range of (S) is 0 to 65535.
- (3) When (S) is a 64-bit floating-point type, its fractional portion is rounded down before conversion is made.
- (4) If (S) is a 16-bit integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

- (1) An operation error will occur if:
 - The (S) data is outside the range 0 to 65535; or
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which converts the data of D0L into an unsigned 16-bit integer value and substitutes the result to #0



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.7.3 Signed 32-bit integer value conversion : LONG

Format	LONG(S)	Number of basic steps	2
--------	---------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into signed 32-bit integer value	32-bit integer type

[Functions]

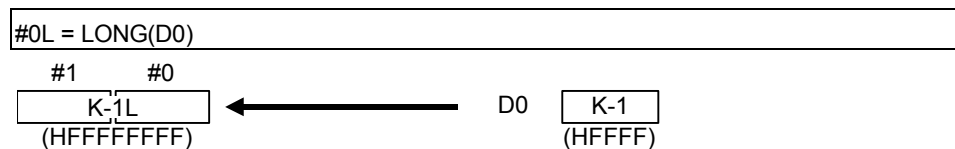
- (1) The data specified with (S) is converted into a signed 32-bit integer value.
- (2) The data range of (S) is -2147483648 to 2147483647.
- (3) When (S) is a 64-bit floating-point type, its fractional portion is rounded down before conversion is made.
- (4) If (S) is a 32-bit integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

- (1) An operation error will occur if:
 - The (S) data is outside the range -2147483648 to 2147483647; or
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which converts the data of D0 into a signed 32-bit integer value and substitutes the result to #0L



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.7.4 Unsigned 32-bit integer value conversion : ULONG

Format	ULONG(S)	Number of basic steps	2
--------	----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into unsigned 32-bit integer value	32-bit integer type

[Functions]

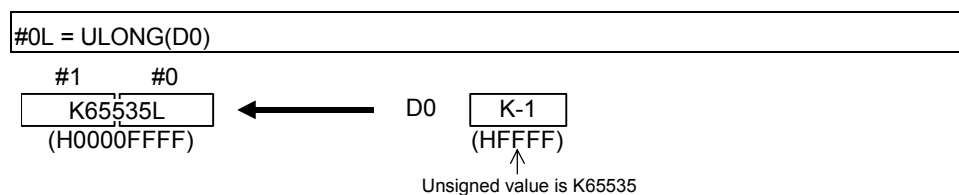
- (1) The data specified with (S) is converted into an unsigned 32-bit integer value.
- (2) The data range of (S) is 0 to 4294967295.
- (3) When (S) is a 64-bit floating-point type, its fractional portion is rounded down before conversion is made.
- (4) If (S) is a 32-bit integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

- (1) An operation error will occur if:
 - The (S) data is outside the range 0 to 4294967295; or
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which converts the data of D0 into an unsigned 32-bit integer value and substitutes the result to #0L



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.7.5 Signed 64-bit floating-point value conversion : FLOAT

Format	FLOAT(S)	Number of basic steps	2
--------	----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into signed 64-bit floating-point value	64-bit floating-point type

[Functions]

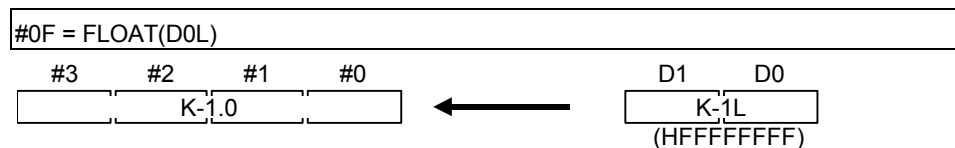
- (1) The data specified with (S) is converted into a signed 64-bit floating-point value.
- (2) If (S) is a 64-bit floating-point type, its value is returned unchanged, with no conversion processing performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which converts the data of D0L into a signed 64-bit floating-point value and substitutes the result to #0F



5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.7.6 Unsigned 64-bit floating-point value conversion : UFLOAT

Format	UFLOAT(S)	Number of basic steps	2
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into unsigned 64-bit floating-point value	64-bit floating-point type

[Functions]

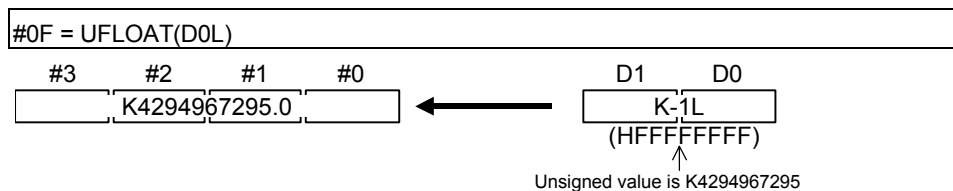
- (1) The data specified with (S) is converted into an unsigned 64-bit floating-point value.
- (2) If (S) is a 64-bit floating-point type, its value is returned unchanged, with no conversion processing performed.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which converts the data of D0L into an unsigned 64-bit floating-point value and substitutes the result to #0F



5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.8 Bit Device Statuses

5.8.1 ON (Normally open contact) : (None)

Format	(S)	Number of basic steps	2
--------	-----	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	○	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Bit device used in bit conditional expression	Logical type (true/false)

[Functions]

- (1) True is returned when the bit device specified with (S) in a bit conditional expression is ON (1), or false is returned when that bit device is OFF (0).

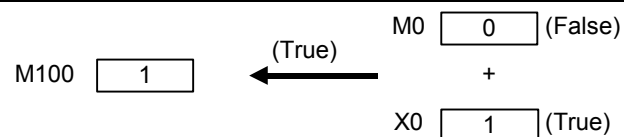
[Errors]

- (1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which sets M100 when either of M0 and X0 is ON (1)

SET M100 = M0 + X0



5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.8.2 OFF (Normally closed contact) : !

Format	!(S)	Number of basic steps	2
--------	------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	○	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Bit device used in bit conditional expression	Logical type (true/false)

[Functions]

- (1) True is returned when the bit device specified with (S) in a bit conditional expression is OFF (0), or false is returned when that bit device is ON (1).

[Errors]

- (1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which resets M100 when M0 is OFF (0)

RST M100 = !M0		
M100	0	← !M0 0 (True)

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.9 Bit Device Controls

5.9.1 Device set : SET

Format	SET(D)=(S)	Number of basic steps	4
--------	------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	<input type="radio"/>	—	—	—	—	—	—	—	—	—	—
(S)	<input type="radio"/>	—	—	—	—	—	—	—	—	<input type="radio"/>	<input type="radio"/>

☐ : Usable

(Note-1) : PX is write-disabled and cannot be used at (D).

(Note-2) : M2001 to M2032 cannot be used at (D).

[Setting data]

Setting data	Description	Data type of result
(D)	Bit data for device set	Bit logical type (true/false)
(S)	Condition data which determines whether device set will be performed or not	

[Functions]

- (1) If the data specified with (S) is true, the bit data specified with (D) is set.
- (2) (S) can be omitted.
At this time, the format is "SET(D)" and device set is made unconditionally.
- (3) When this instruction is set as a transition condition in the last block of a transition program, whether the data specified with (S) is true or false is returned as logical type data. In this case, (S) cannot be omitted.

[Errors]

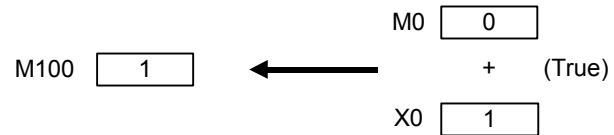
- (1) An operation error will occur if:
 - (D) or (S) is an indirectly specified device and its device No. is outside the range.

5 OPERATION CONTROL PROGRAMS

[Program examples]

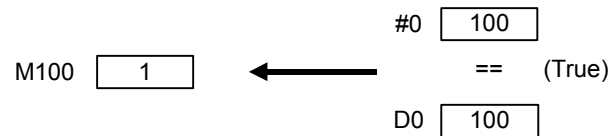
- (1) Program which sets M100 when either of M0 and X0 is 1

SET M100 = M0 + X0



- (2) Program which sets M100 when #0 is equal to D0

SET M100 = #0 == D0



- (3) Program which sets Y0 unconditionally

SET Y0



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.9.2 Device reset : RST

Format	RST(D)=(S)	Number of basic steps	4
--------	------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	<input type="radio"/>	—	—	—	—	—	—	—	—	—	—
(S)	<input type="radio"/>	—	—	—	—	—	—	—	—	<input type="radio"/>	<input type="radio"/>

☐ : Usable

(Note-1) : PX is write-disabled and cannot be used at (D).

(Note-2) : M2001 to M2032 cannot be used at (D).

[Setting data]

Setting data	Description	Data type of result
(D)	Bit data for device reset	Bit logical type (true/false)
(S)	Condition data which determines whether device reset will be performed or not	

[Functions]

- (1) If the data specified with (S) is true, the bit data specified with (D) is reset.
- (2) (S) can be omitted.
At this time, the format is "RST(D)" and device reset is made unconditionally.
- (3) When this instruction is set as a transition condition in the last block of a transition program, whether the data specified with (S) is true or false is returned as logical type data. In this case, (S) cannot be omitted.

[Errors]

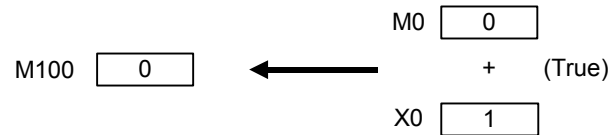
- (1) An operation error will occur if:
 - (D) or (S) is an indirectly specified device and its device No. is outside the range.

5 OPERATION CONTROL PROGRAMS

[Program examples]

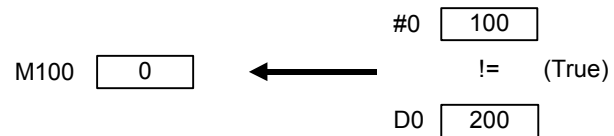
- (1) Program which resets M100 when either of M0 and X0 is 1

RST M100 = M0 + X0



- (2) Program which resets M100 when #0 is equal to D0

RST M100 = #0 != D0



- (3) Program which resets Y0 unconditionally

RST Y0



5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.9.3 Device output : DOUT

Format	DOUT(D), (S)	Number of basic steps	4
--------	--------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	—	—	—	—	—	—	—	—	—	—
(S)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

(Note-1) : PX and special relay cannot be used at (D).

(Note-2) : Range including M2000 to M2127 cannot be used at (D).

[Setting data]

Setting data	Description	Data type of result
(D)	Output destination bit data	Batch bit
(S)	Output source data	

[Functions]

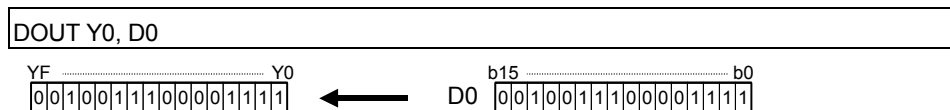
- (1) The data specified with (S) is output to the bit data specified with (D).
- (2) Specify a multiple of 16 as the device No. of the bit data specified with (D).
- (3) If the type of (S) is a 16-bit integer type, 16 points of the (S) data, starting at the least significant bit, are output in order to the bit devices headed by the one specified with (D).
- (4) If the type of (S) is a 32-bit integer type, 32 points of the (S) data, starting at the least significant bit, are output in order to the bit devices headed by the one specified with (D).

[Errors]

- (1) An operation error will occur if:
 - (D) or (S) is an indirectly specified device and its device No. is outside the range.
 - (D) is an indirectly specified device and its device No. is not a multiple of 16.

[Program examples]

- (1) Program which outputs the data of D0 to Y0-YF



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.9.4 Device input : DIN

Format	DIN(D), (S)	Number of basic steps	4
--------	-------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	—	<input type="radio"/>	<input type="radio"/>	—	—	—	—	—	—	—	—
(S)	<input type="radio"/>	—	—	—	—	—	—	—	—	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Input destination data	Data type of (D) (Integer type)
(S)	Input source bit data	

[Functions]

- (1) The bit data specified with (S) is input to the data specified with (D).
- (2) Specify a multiple of 16 as the device No. of the bit data specified with (S).
- (3) If the type of (D) is a 16-bit integer type, 16 points of the (D) data, starting at the least significant bit, are input in order to the bit devices headed by the one specified with (S).
- (4) If the type of (D) is a 32-bit integer type, 32 points of the (D) data, starting at the least significant bit, are input in order to the bit devices headed by the one specified with (S).

[Errors]

- (1) An operation error will occur if:
 - (D) or (S) is an indirectly specified device and its device No. is outside the range.
 - (S) is an indirectly specified device and its device No. is not a multiple of 16.

[Program examples]

- (1) Program which inputs the data of X0-XF to D0



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.9.5 Bit device output : OUT

Format	OUT(D)=(S)	Number of basic steps	4
--------	------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	<input type="radio"/>	—	—	—	—	—	—	—	—	—	—
(S)	<input type="radio"/>	—	—	—	—	—	—	—	—	<input type="radio"/>	<input type="radio"/>

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Bit device for device output	Bit logical type (true/false)
(S)	Condition data which determines device output	

[Functions]

- (1) If the data specified with (S) is true, the bit data specified with (D) is set, and if the data specified with (S) is false, the bit data specified with (D) is reset.
- (2) When this instruction is set as a transition condition in the last block of a transient program, whether the data specified with (S) is true or false is returned as logical type data.
- (3) In this case, (S) cannot be omitted.

[Errors]

- (1) An operation error will occur if:
 - (D) or (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which sets M100 when M0 is ON (1) and program which resets M100 when M0 is OFF (0)

```
OUT M100 = M0
```

- (2) Program which sets M100 when M0 and M1 are both on and resets M100 except it

```
OUT M100 = M0 * M1
```

- (3) Program which sets M100 when D0 is equal to D2000 and resets M100 when D is not equal to D2000

```
OUT M100 = (D0 == D2000)
```

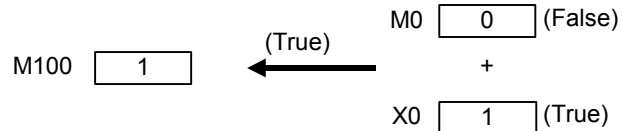
--	--

○ : Usable[illegible]

- (1) Whether the logical type date specified with (C) is true or false is returned

- (1) An operation error will occur if:

- (4) Program which sets M100 when either of M0 and Y0 is ON (4)



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.10.2 Logical negation : !

Format	! (S)
--------	-------

Number of basic steps	2
-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	<input type="radio"/>	—	—	—	—	—	—	—	—	<input type="radio"/>	<input type="radio"/>

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be logically negated	Logical type (true/false)

[Functions]

- (1) The data specified with (S) is logically negated.

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which sets M100 when "either of M0 and X0 is not ON (1)" (when M0 and X0 are both OFF (0))

SET M100 = ! (M0 + X0)									
				M0	<input type="text" value="0"/>	(False)			
						+			
				X0	<input type="text" value="0"/>	(False)			
M100	<input type="text" value="1"/>					(True) ! (False)			

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.10.3 Logical AND : *

Format	(S1)*(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	<input type="radio"/>	—	—	—	—	—	—	—	—	<input type="radio"/>	<input type="radio"/>
(S2)	<input type="radio"/>	—	—	—	—	—	—	—	—	<input type="radio"/>	<input type="radio"/>

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be ANDed	Logical type (true/false)
(S2)		

[Functions]

- (1) The data specified with (S1) and the data specified with (S2) are ANDed.

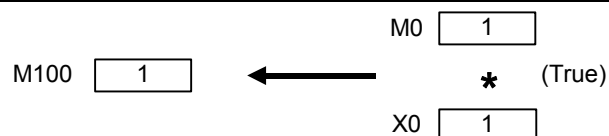
[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which sets M100 when M0 and X0 are both 1

SET M100 = M0 * X0



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.10.4 Logical OR : +

Format	(S1)+(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	<input type="radio"/>	—	—	—	—	—	—	—	—	<input type="radio"/>	<input type="radio"/>
(S2)	<input type="radio"/>	—	—	—	—	—	—	—	—	<input type="radio"/>	<input type="radio"/>

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be ORed	Logical type (true/false)
(S2)		

[Functions]

- (1) The data specified with (S1) and the data specified with (S2) are ORed.

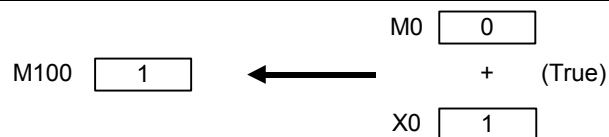
[Errors]

- (1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which sets M100 when either of M0 and X0 is 1

SET M100 = M0 + X0



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.11 Comparison Operations

5.11.1 Equal to : ==

Format	(S1)==(S2)	Number of basic steps	4
--------	------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

- The data specified with (S1) and the data specified with (S2) are compared, and the result is true if they are equal.
- When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

- An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- Program which compares whether #0 and D0 are equal or not

#0 == D0		
#0	<input type="text" value="100"/>	
(True) ←	==	
D0	<input type="text" value="100"/>	

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.11.2 Not equal to : !=

Format	(S1)!(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

- (1) The data specified with (S1) and the data specified with (S2) are compared, and the result is true if they are not equal.
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which compares whether #0 and D0 are unequal or not

#0 != D0		
#0	<input type="text" value="100"/>	
(True) ←	!=	
D0	<input type="text" value="20"/>	

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.11.3 Less than : <

Format	(S1)<(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

- (1) The result is true if the data specified with (S1) is less than the data specified with (S2).
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which compares whether #0 is less than D0 or not

#0 < D0		
#0	<input type="text" value="10"/>	
(True) ←	<	
D0	<input type="text" value="20"/>	

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.11.4 Less than or equal to: <=

Format	(S1)<=(S2)	Number of basic steps	4
--------	------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

- (1) The result is true if the data specified with (S1) is less than or equal to the data specified with (S2).
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which compares whether #0 is less than or equal to D0 or not

#0 <= D0		
#0	<input type="text" value="10"/>	
(True) ←	<=	
D0	<input type="text" value="20"/>	

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.11.5 More than : >

Format	(S1)>(S2)	Number of basic steps	4
--------	-----------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

- (1) The result is true if the data specified with (S1) is greater than the data specified with (S2).
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which compares whether #0 is greater than D0 or not

#0 > D0		
#0	400	
(True) ←	>	
D0	20	

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.11.6 More than or equal to: >=

Format	(S1)>=(S2)	Number of basic steps	4
--------	------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

- (1) The result is true if the data specified with (S1) is greater than or equal to the data specified with (S2).
- (2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

- (1) An operation error will occur if:
 - (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

- (1) Program which compares whether #0 is greater than or equal to D0 or not

#0 >= D0		
#0	<input type="text" value="400"/>	
(True) ←	>=	
D0	<input type="text" value="20"/>	

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.12 Motion-Dedicated Functions (CHGV, CHGT)

5.12.1 Speed change request : CHGV

Format	CHGV((S1), (S2))	Number of basic steps	4
--------	------------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	—	—	—	—	<input type="radio"/>	—	—	—	—	—
(S2)	—	<input type="radio"/>	<input type="radio"/>	—	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Axis No. to which speed change request will be given	—
(S2)	Specified speed	

[Functions]

- (1) A speed change is made in the following procedure.
 - (a) The speed changing flag (M2061 to M2092) correspond to the axis specified with (S1) is turned ON.
 - (b) The speed of the axis specified with (S1) is changed to the speed specified with (S2).
 - (c) The speed changing flag is turned OFF.
- (2) The axis No. that may be set at (S1) is within the following range.

Q172DCPU	Q173DCPU
1 to 8	1 to 32

For interpolation control, set any one of the interpolation axes. When linear interpolation control is exercised, a speed change varies as described below with the positioning speed designation method set in the servo program.

Positioning speed designation method	Operation
Vector speed designation	Speed change is made so that the vector speed becomes the speed specified with (S2).
Longest axis designation	Speed change is made so that the longest axis speed becomes the speed specified with (S2).
Reference axis speed designation	Speed change is made so that the reference axis speed becomes the speed specified with (S2).

5 OPERATION CONTROL PROGRAMS

- (3) Operation varies with the sign of the specified speed set at (S2).

Sign of specified speed	Operation
Positive	Speed change
0	Temporary stop
Negative	Return

- (4) The specified speed that may be set at (S2) is within the following range.

(a) Real mode

	mm		inch		degree		PLS	
	Setting range	Unit	Setting range	Unit	Setting range	Unit	Setting range	Unit
Speed change request	0 to 600000000	$\times 10^{-2}$ mm/min	0 to 600000000	$\times 10^{-3}$ inch/min	0 to 2147483647	$\times 10^{-3}$ degree/min (Note-1)	0 to 2147483647	PLS/s
Return request	-1 to -600000000	$\times 10^{-2}$ mm/min	-1 to -600000000	$\times 10^{-3}$ inch/min	-1 to -2147483647	$\times 10^{-3}$ degree/min (Note-1)	-1 to -2147483647	PLS/s

(Note-1) : When the "speed control 10 \times multiplier setting for degree axis" is set to "valid" in the fixed parameter, the unit is " $\times 10^{-2}$ [degree/min]".

(b) Virtual mode

	PLS	
	Setting range	Unit
Speed change request	0 to 2147483647	PLS/s
Return request	-1 to -2147483647	PLS/s

- (5) The speed changed by CHGV instruction is effective only on the servo program during starting.
- (6) The speed change does not executed for the axis specified with (S1) during deceleration stop.

5 OPERATION CONTROL PROGRAMS

- (7) By specifying a negative speed and making a speed change request during the start, allows the axis to start deceleration at that point and return in the opposite direction upon completion of deceleration.

The following operations by the servo instruction are shown below.

Control mode	Servo instruction	Operation
Linear control	<input type="checkbox"/> ABS-1 <input type="checkbox"/> INC-1	On completion of deceleration, the axis reverses its moving direction, returns to the positioning starting point at the absolute value of the specified speed, and stops (waits) there.
	<input type="checkbox"/> ABS-2 <input type="checkbox"/> INC-2	
	<input type="checkbox"/> ABS-3 <input type="checkbox"/> INC-3	
	<input type="checkbox"/> ABS-4 <input type="checkbox"/> INC-4	
Circular interpolation control	<input type="checkbox"/> ABS circular <input type="checkbox"/> INC circular	For circular interpolation, the axis returns in the circular path.
Fixed-pitch feed	<input type="checkbox"/> FEED-1 <input type="checkbox"/> FEED-2 <input type="checkbox"/> FEED-3	
Constant-speed control	<input type="checkbox"/> CPSTART1 <input type="checkbox"/> CPSTART2	On completion of deceleration, the axis reverses its moving direction, returns to the preceding point at the absolute value of the specified speed, and stops (waits) there.
	<input type="checkbox"/> CPSTART3 <input type="checkbox"/> CPSTART4	
Speed control (I)	<input type="checkbox"/> VF <input type="checkbox"/> VR	On completion of deceleration, the axis reverses its moving direction at the absolute value of the specified speed. The axis does not stop until a stop instruction is input.
Speed control (II)	<input type="checkbox"/> VVF <input type="checkbox"/> VVR	
Speed/position control	<input type="checkbox"/> VPF <input type="checkbox"/> VPR <input type="checkbox"/> VPSTART	The axis cannot return. The speed change request is regarded as a normal speed change request. Minor error [305] ^(Note) will occur and the axis will be controlled at the speed limit value.
Position follow-up control	<input type="checkbox"/> PFSTART	
Speed control with fixed position stop	<input type="checkbox"/> PVF <input type="checkbox"/> PVR	
Speed switching control	<input type="checkbox"/> VSTART	
JOG operation		A speed change cannot be made. Minor error [310] ^(Note) will occur.
High-speed oscillation	<input type="checkbox"/> OSC	
Home position return	<input type="checkbox"/> ZERO	A speed change cannot be made. Minor error [301] ^(Note) will occur.

(Note) : Minor error [301] : A speed change was made during home position return.

Minor error [305] : The setting speed is outside the range of 0 to speed limit value.

Minor error [310] : A speed change was made during high-speed oscillation.

[Controls]

- (a) If a speed change is made to a negative speed, control is executed with the control mode during the start as indicated in the above table.
- (b) The returning command speed is the absolute value of a new speed.
- (c) When the axis is waiting at the return position
 - 1) Signal states (n : Axis No., m : Axis No. -1)
 - Start accept (M2000+n) ON
(unchanged from before execution of CHGV instruction)
 - Positioning start completion (M2400+20m) ON
(unchanged from before execution of CHGV instruction)
 - Positioning completion (M2401+20m) OFF
 - In-position (M2402+20m) ON
 - Command in-position (M2403+20m) OFF
 - Speed change "0" accepting flag (M2240+m) ON

5 OPERATION CONTROL PROGRAMS

- 2) Make a speed change to a positive speed for a restart.
- 3) Turn on the stop command to end the positioning.
- 4) A negative speed change made again will be ignored.

- (d) While the axis is reversion in the speed control mode
- 1) Make a speed change to a positive speed to change the travel direction again.
 - 2) Turn ON the stop command to make a stop.
 - 3) A speed change is made in the opposite direction if a negative speed change is made again.

[Errors]

- (1) An operation error will occur and a speed change will not be made if:
- The specified axis No. of (S1) is outside the range.
 - (S2) is an indirectly specified device and its device No. is outside the range.
- (2) A minor error will occur and a speed change will not be made if:
- The axis specified with (S1) is home position return. (Minor error: 301).

POINT

If the speed change is executed for the axis specified with (S1) during deceleration, the speed change is ignored. An error will not occur in this case

- (3) A minor error will occur and the axis to be controlled at the speed limit value if:
- The absolute value of the speed specified with (S2) is greater than the speed limit value. (Minor error: 305)

POINT

If the absolute value of a negative new speed is higher than the speed specified with the servo program during constant-speed control, return control is exercised at the speed specified in the program (speed clamp control for a speed change during constant-speed control). At this time, an error will not occur.
--

5 OPERATION CONTROL PROGRAMS

[Program examples]

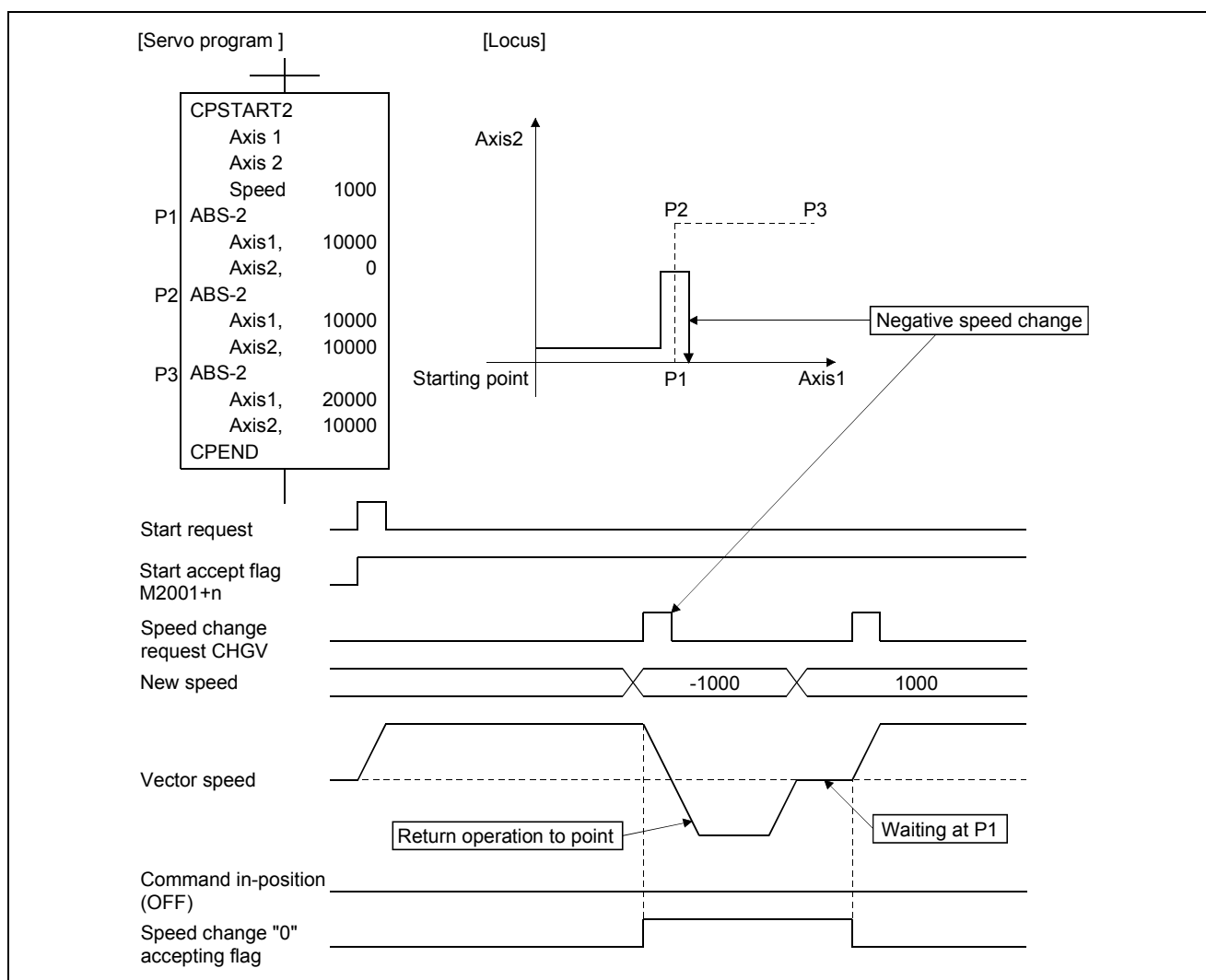
- (1) Program which changes the positioning speed of axis 2

```
CHGV(K2,K10)
```

- (2) Return program which changes the positioning speed of axis 1 to a negative value

```
CHGV(K1,K-1000)
```

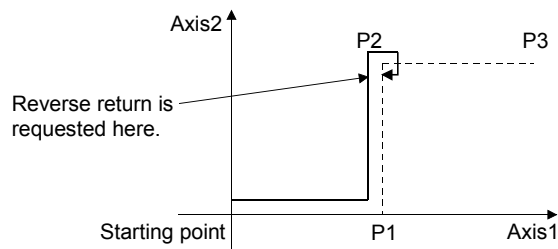
The following operation will be performed when a return request is made in constant-speed control.



If a speed change to a negative speed is made during execution of positioning to P2 as shown above, the axis returns to P1 along the program specified locus and waits at P1.

POINT

- Precautions at speed change
 - (1) A speed change may be invalid if the speed change is executed until the "positioning start complete signal" status changes to ON at servo program start request . When making a speed change at almost the same timing as a start, create a program to execute speed change after the "positioning start complete signal" has turned on.
 - (2) When the reverse return is requested during stop in the state of FIN waiting using the M-code FIN signal wait function in constant-speed control, it will be ignored.
 - (3) In the above example, if reverse return is requested before P2 and the axis passes through P2 during deceleration, it return to P2.
 - (4) There will be a delay of time equivalent to an operation cycle at the maximum in the response time from when the CHGV instruction is executed until the speed begins to change actually.



5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.12.2 Torque limit value change request : CHGT

Format	CHGT((S1), (S2))	Number of basic steps	4
--------	------------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	—	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Axis No. to which torque limit value change request will be given	—
(S2)	Specified torque limit value	

[Functions]

- (1) The torque limit value of the axis specified with (S1) is changed to the torque limit value axis specified with (S2).
- (2) In the real mode, any axis that has completed a servo startup can be changed in torque limit value any time, independently of the status, starting, stopping, servo ON or servo OFF.
- (3) The axis No. that may be set at (S1) is within the following range.

Q172DCPU	Q173DCPU
1 to 8	1 to 32

- (4) The torque limit value that may be set at (S2) is within the range 1 to 1000[%].
- (5) The torque limit value specified here and the one specified in the servo program have the following relationships.

At start

At a normal start, the torque limit value is given to the servo of the start axis according to "P. torque" set in the servo program or the "torque limit value" of the specified parameter block.

For an interpolation start, the torque limit value is given to the number of axes to be interpolated.



Executing the CHGT instruction gives the preset torque limit value to only the specified axis.



Thereafter, the torque limit value given to the servo at a servo program start or JOG start is made valid only when it is lower than the torque limit value specified in CHGT.

This torque limit value clamp processing is performed per axis.

During start

- (a) If the following torque limit value has been set, it will not be changed to higher than the torque limit value specified in the CHGT instruction.
 - Torque limit value at a midway point in constant-speed control or speed switching control
 - Torque limit value at the point of switching to position control in speed/ position changing control
 - Torque limit value in speed control
- (b) The CHGT instruction accepts a torque limit value which is higher than the torque limit value set in the servo program or parameter block.
- (6) The torque limit value changed by CHGT instruction is effective only during power supply is on.

[Errors]

- (1) An operation error will occur and a torque limit value change will not be made if:
 - The specified axis No. at (S1) is outside the range; or
 - (S2) is an indirectly specified device and its device No. is outside the range.
- (2) A minor error will occur and a torque limit value change will not be made if:
 - The torque limit value specified with (S2) is outside the range 1 to 1000[%] (Minor error: 311); or
 - The CHGT instruction is executed for any axis that has not yet been started (Minor error: 312).

[Program examples]

- (1) Program which changes the torque limit value of axis 2

CHGT(K2,K10)

POINT

- (1) CHGT instruction is invalid (ignored) during the virtual mode. When changing the torque limit value during operation in the virtual mode, set the "torque limit value setting device" in the output module parameter of the mechanical system program.
- (2) There will be a delay of time equivalent to an operation cycle at the maximum in the response time from when the CHGT instruction is executed until the torque limit value is changed actually.

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.13 Other Instructions

5.13.1 Event task enable : EI

Format	EI	Number of basic steps	1
--------	----	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
—	—	—	—	—	—	—	—	—	—	—	—

☐ : Usable

[Setting data]

There are no setting data.

[Functions]

- (1) The execution of an event task is enabled.
- (2) This instruction is usable with a normal task only.

[Errors]

- (1) An operation error will occur if:
 - This instruction is used with other than a normal task.

[Program examples]

- (1) Enables the execution of an event task.

EI

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.13.2 Event task disable : DI

Format	DI	Number of basic steps	1
--------	----	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
—	—	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

There are no setting data.

[Functions]

- (1) The execution of an event task is disabled.
- (2) If an external interrupt or PLC interrupt occurs after execution of the DI instruction, the corresponding event task is executed once at the execution of the EI instruction. (If two or more external interrupts or PLC interrupts occur during DI, the corresponding event task is executed only once at the execution of the EI instruction.)
- (3) During DI, a fixed-cycle event task is not executed.
- (4) The execution of an NMI task cannot be disabled.
- (5) The DI status is established at power-on or reset of the Multiple CPU system.

[Errors]

- (1) An operation error will occur if:
 - This instruction is used with other than a normal task.

[Program examples]

- (1) Program which disables the execution of an event task.

DI

5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.13.3 No operation : NOP

Format	NOP	Number of basic steps	1
--------	-----	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
—	—	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

There are no setting data.

[Functions]

- (1) This is a no-operation instruction and does not affect the preceding operations.

[Errors]

- (1) There are no operation errors for no operation: NOP.

5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.13.4 Block transfer : BMOV

Format	BMOV(D), (S), (n)	Number of basic steps	6
--------	-------------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	<input type="radio"/>	<input type="radio"/>	—	—	—	—	<input type="radio"/>	—	—	—	—
(S)	<input type="radio"/>	<input type="radio"/>	—	—	—	—	<input type="radio"/>	—	—	—	—
(n)	—	<input type="radio"/>	—	—	—	<input type="radio"/>	—	—	—	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Transfer destination device starting No.	—
(S)	Transfer source device starting No.	
(n)	Number of words to be transferred	

[Functions]

- (1) The contents for n words from device specified with (S) are batch-transferred to the n words from device specified with (D).
- (2) Data can be transferred if the devices of the transfer source and destination overlap.
Data are transferred from devices, starting with the one at (S), for transfer of data from devices of larger numbers to those of smaller numbers, or starting with the one at (S)+(n-1) for transfer of data from devices of smaller numbers to those of larger numbers.
- (3) Specifying Nn (cam No.) at (D) or (S) enables batch-transfer of cam data.
In the Motion controller, the cam data of same cam No. must already have been registered.
The number of transferred words specified with (n) should match the resolution of the specified cam No..

At cam data write

The cam data storage area is rewritten.

- Transfer of data to the cam data area is also executed during cam operation.
Be careful not to perform write while operation is being performed with the same cam No..

At cam data read

The cam data storage area is rewritten.

- The cam data in the currently set status are read.

5 OPERATION CONTROL PROGRAMS

(4) The word devices that may be set at (D), (S) and (n) are shown below.

Setting data	Word devices ^(Note-2)					Bit devices ^{(Note-2), (Note-3)}							Cam No. specification
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn	Nn ^(Note-1)
(D)	○	○	—	○	○	○	—	○	○	—	○ ^(Note-4)	○ ^(Note-4)	○
(S)	○	○	○	○	○	○	—	○	○	○	○ ^(Note-4)	○ ^(Note-4)	○
(n)	○	○	—	○	○	—	—	—	—	—	—	—	—

(Note-1) : "Nn" indicates the cam No..

(Note-2) : The device No. cannot be specified indirectly.

(Note-3) : Specify a multiple of 16 as the device number of bit data.

(Note-4) : PX/PY cannot be set.

(5) The cam No. that may be set as "Nn" is within the following range.

Q173DPU/Q172DCPU
1 to 64
101 to 164
201 to 264
301 to 364

[Errors]

(1) An operation error will occur if:

- The cam data of cam No. specified with (D) or (S) are not yet registered to the Motion controller
- The resolution of cam No. specified with (D) or (S) differs from the number of transferred words specified with (n)
- (S) to (S)+(n-1) is outside the device range
- (D) to (D)+(n-1) is outside the device range
- (n) is 0 or a negative number
- PX/PY is set in (S) to (S)+(n-1)
- PX/PY is set in (D) to (D)+(n-1)

} when (n) specified is a word device

(2) When conversion is made in program editing of MT Developer, an error will occur if:

- (S) to (S)+(n-1) is outside the device range
- (D) to (D)+(n-1) is outside the device range
- (n) is 0 or a negative number
- PX/PY is set in (S) to (S) + (n-1)
- PX/PY is set in (D) to (D) + (n-1)
- (S) is a bit device and the device number is not a multiple of 16
- (D) is a bit device and the device number is not a multiple of 16

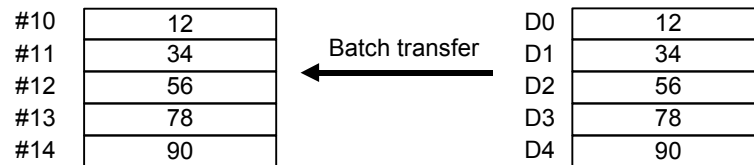
} when (n) specified is a constant

5 OPERATION CONTROL PROGRAMS

[Program examples]

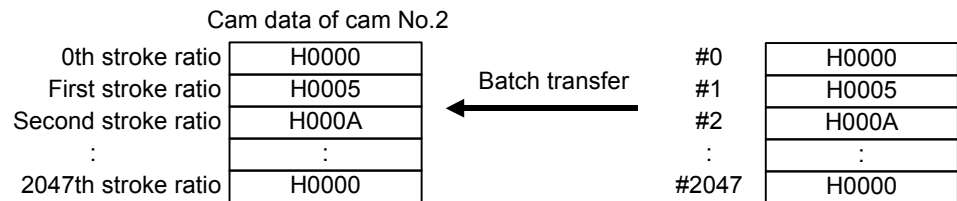
- (1) Program which batch-transfers a contents for 5 words from D0 to all data for 5 words from #10

```
BMOV #10,D0,K5
```



- (2) Program which batch-transfers a contents for 2048 words from #0 to the data area of cam No.2 (resolution 2048)

```
BMOV N2,#0,K2048
```

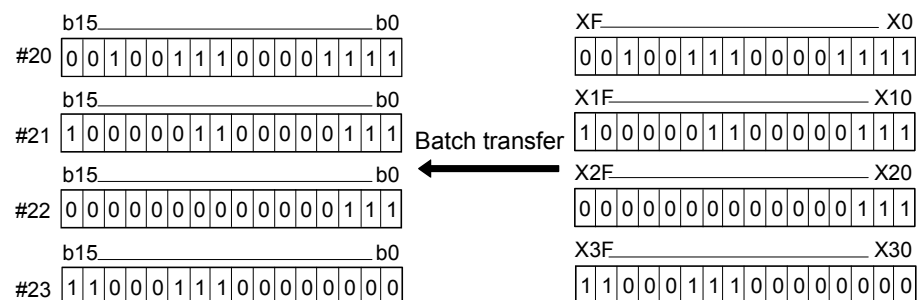


POINT

Cam stroke ratio is set within 0 to 7FFFH.

- (3) Program which batch-transfers a contents for 4 words from X0 to all data for 4 words from #20

```
BMOV #20, X0, K4
```



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.13.5 Same data block transfer : FMOV

Format	FMOV(D), (S), (n)	Number of basic steps	6
--------	-------------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	<input type="radio"/>	<input type="radio"/>	—	—	—	—	<input type="radio"/>	—	—	—	—
(S)	<input type="radio"/>	<input type="radio"/>	—	—	—	<input type="radio"/>	—	—	—	—	—
(n)	—	<input type="radio"/>	—	—	—	<input type="radio"/>	—	—	—	—	—

☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Transfer destination device starting No.	—
(S)	Device No. which transfer data or data to be transferred are stored.	
(n)	Number of words to be transferred	

[Functions]

- (1) The data specified with (S) or contents of device are transferred a part for (n)words of data to the device specified with (D).

- (2) The devices that may be set at (D), (S) and (n) are shown below.

Setting data	Word devices ^(Note-1)					Bit devices ^{(Note-1), (Note-2)}						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D)	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/> ^(Note-3)	<input type="radio"/> ^(Note-3)
(S)	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/> ^(Note-3)	<input type="radio"/> ^(Note-3)
(n)	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	—	—	—	—	—	—	—

(Note-1) : The device No. cannot be specified indirectly.

(Note-2) : Specify a multiple of 16 as the device number of bit data.

(Note-3) : PX/PY cannot be set.

[Errors]

- (1) An operation error will occur if:
- (D) to (D)+(n-1) is outside the device range
 - (n) is 0 or a negative number
 - PX/PY is set in (D) to (D)+(n-1)
- } When (n) specified is a word device

5 OPERATION CONTROL PROGRAMS

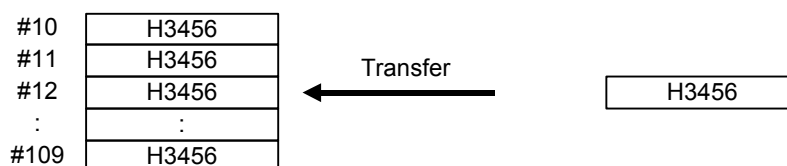
(2) When conversion is made in program editing of MT Developer, an error will occur if:

- (D) to (D)+(n-1) is outside the device range
 - (S) is outside the device range
 - (n) is 0 or a negative number
 - PX/PY is set in (S)
 - PX/PY is set in (D) to (D) + (n-1)
 - (S) is a bit device and the device number is not a multiple of 16
 - (D) is a bit device and the device number is not a multiple of 16
- } When (n) specified is a constant

[Program examples]

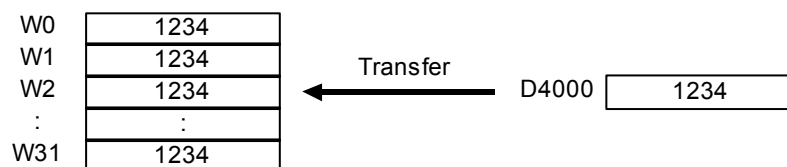
(1) Program which sets 3456H to all data for 100 words from #10

```
FMOV #10,H3456,K100
```



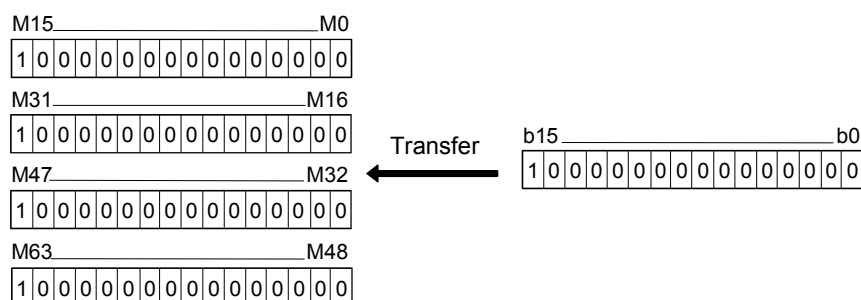
(2) Program which sets a content of D4000 to all data for 50 words from W0

```
FMOV W0,D4000,K50
```



(3) Program which sets 8000H to all data for 4 words from M0

```
FMOV M0, H8000, K4
```



5 OPERATION CONTROL PROGRAMS

F/FS	G
<input type="radio"/>	<input type="radio"/>

5.13.6 Write device data to CPU shared memory of the self CPU: MULTW

Format	MULTW(D), (S), (n), (D1)	Number of basic steps	8
--------	--------------------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	—	<input type="radio"/>	—	—	—	<input type="radio"/>	—	—	—	—	—
(S)	<input type="radio"/>	<input type="radio"/>	—	—	—	—	—	—	—	—	—
(n)	—	<input type="radio"/>	—	—	—	<input type="radio"/>	—	—	—	—	—
(D1)	<input type="radio"/>	—	—	—	—	—	—	—	—	—	—

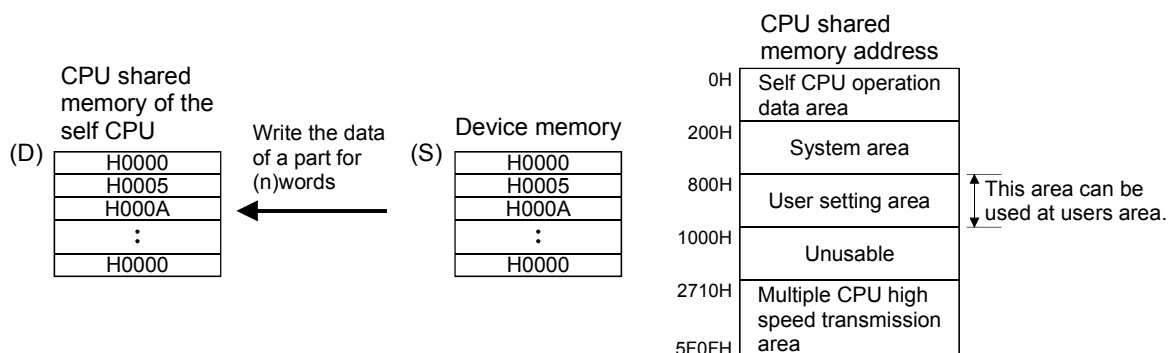
☐ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	The CPU shared memory address of self CPU of the writing destination device. (800H to FFFH)	—
(S)	Start device No. which writing data are stored.	
(n)	Number of words to be written (1 to 256)	
(D1)	Self CPU device is made to turn on by the writing completion.	

[Functions]

- (1) A part for (n)words of data since the device specified with (S) of the self CPU module are written to since the CPU shared memory address specified with (D) of the self CPU module. After writing completion of the device data, the complete bit device specified with (D1) turns on.



- (2) Do resetting of the complete bit device by the user program.

5 OPERATION CONTROL PROGRAMS

- (3) Another MULTW instruction cannot be processed until MULTW instruction is executed and a complete bit device is turned on. When MULTW instruction was executed again before MULTW instruction is executed and complete bit device is turned on, the MULTW instruction executed later becomes an error.

- (4) The word devices that may be set at (D), (S) (n) and (D1) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D)	○	○	—	○	○	—	—	—	—	—	—	—
(S)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(n)	○	○	—	○	○	—	—	—	—	—	—	—
(D1)	—	—	—	—	—	○	○	○	○	—	○ (Note-4)	○ (Note-4)

(Note-1) : The device No. cannot be specified indirectly.

(Note-2) : Specify a multiple of 16 as the device number of bit data.

(Note-3) : PX and PY cannot be set.

(Note-4) : PX can be set. PY cannot be set.

- (5) Adjust an executive task, the number of transfer word referring to the operation processing time so that this instruction may not obstruct the execution of the motion operation because processing time becomes long in argument to the number of words (n) to be written.

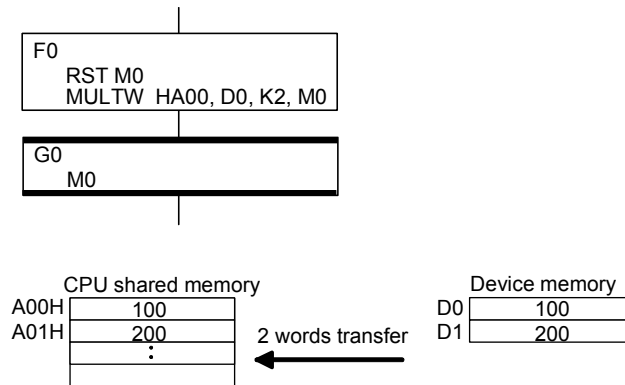
[Errors]

- (1) An operation error will occur if:
- Number of words (n) to be written is outside the range of 1 to 256.
 - The CPU shared memory address (D) of self CPU of the writing destination device is outside the range (800H to FFFH) of the CPU shared memory address.
 - The CPU shared memory address (D) of self CPU of the writing destination device + number of words (n) to be written is outside the range (800H to FFFH) of the CPU shared memory address.
 - Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range.
 - MULTW instruction was executed again before MULTW instruction is executed and complete bit device is turned on.
 - (D1) is a write-disabled device.
 - (S) is a bit device and device number is not a multiple of 16.
 - PX/PY is set in (S) to (S)+(n-1).

5 OPERATION CONTROL PROGRAMS

[Program examples]

- (1) 2 words from D0 is written in the CPU shared memory to since A00H, and transits to next step after confirmation of writing completion.



5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.13.7 Read device data from CPU shared memory of the other CPU: MULTR

Format	MULTR(D), (S1), (S2), (n)	Number of basic steps	7
--------	---------------------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	○	—	—	—	—	—	—	—	—	—
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

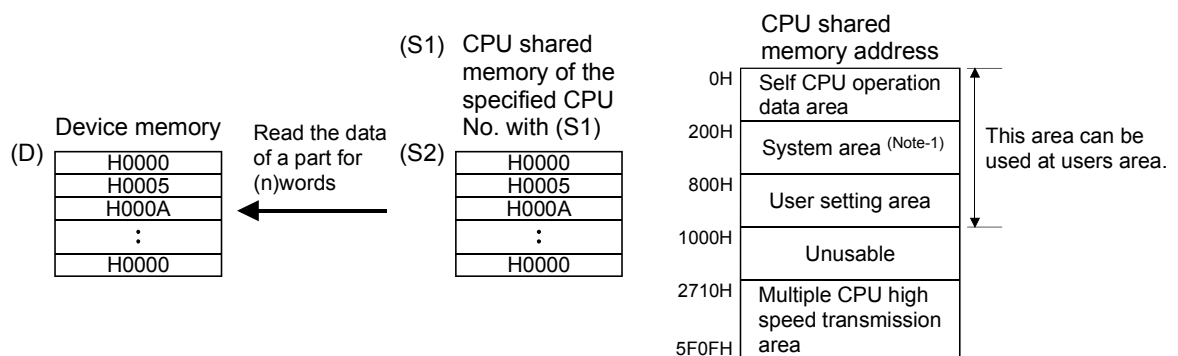
○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Start device No. which stores the reading data.	—
(S1)	First I/O No. of the PLC CPU/Motion CPU which it will be read. (CPU No.1 : 3E0H, CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H)	
(S2)	The CPU shared memory first address of the data which it will be read. (000H to FFFH)	
(n)	Number of words to be read (1 to 256)	

[Functions]

- (1) A part for (n)words of data of the target CPU specified with (S1) are read from the address specified with (S2) of the CPU shared memory, and are stored since the device specified with (S2).



(Note-1): This area cannot be read when the target CPU is self CPU.

5 OPERATION CONTROL PROGRAMS

(2) The word devices that may be set at (D), (S), (n) and (D1) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(S)	○	○	—	○	○	—	—	—	—	—	—	—
(n)	○	○	—	○	○	—	—	—	—	—	—	—
(D1)	○	○	—	○	○	—	—	—	—	—	—	—

(Note-1) : The device No. cannot be specified indirectly.

(Note-2) : Specify a multiple of 16 as the device number of bit data.

(Note-3) : PX and PY cannot be set.

- (3) When data are read normally from the target CPU specified with (S1), the reading complete flag SM528 to SM531 (CPU No.1 : SM528, CPU No.2 : SM529, CPU No.3 : SM530, CPU No.4 : SM531) corresponding to the target CPU turns on. If data cannot be read normally, the reading complete flag of the target CPU does not turn on.
- (4) Adjust an executive task, the number of transfer word referring to the operation processing time so that this instruction may not obstruct the execution of the motion operation because processing time becomes long in argument to the Number of words (n) to be read.
- (5) When multiple MULTR instructions are executed to the same CPU simultaneously, the reading complete flag SM528 to SM531 turns on/as a result of MULTR that it is executed at the end.
- (6) Reset the reading complete flag (SM528 to SM531) using the user program.

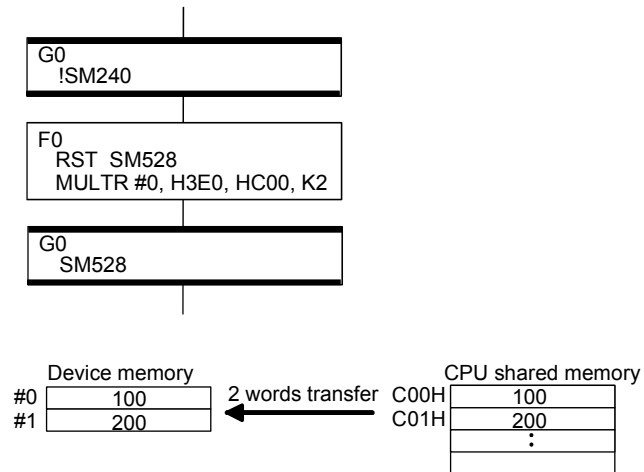
[Errors]

- (1) An operation error will occur if:
 - Number of words (n) to be read is outside the range of 1 to 256.
 - The CPU shared memory first address (S2) of the data which it will be read is outside the range (000H to FFFH) of the CPU shared memory address.
 - The CPU shared memory first address (S2) of the data which it will be read + number of words (n) to be read is outside the range (000H to FFFH) of the CPU shared memory address.
 - Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range.
 - Except 3E0H/3E1H/3E2H/3E3H is set at (S1).
 - The CPU which reads is resetting.
 - The errors are detected in the CPU which read.
 - (D) is a bit device and device number is not a multiple of 16.
 - PX/PY is set in (D) to (D)+(n-1).

5 OPERATION CONTROL PROGRAMS

[Program examples]

- (1) It checks that a CPU No.1 is not resetting, 2 words is read to since #0 from the CPU shared memory C00H of CPU No.1, and transits to next step after reading completion.



5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.13.8 Write device data to intelligent function module : TO

Format	TO(D1), (D2), (S), (n)
--------	------------------------

Number of basic steps	7
-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D1)	—	○	—	—	—	○	—	—	—	—	—
(D2)	—	○	—	—	—	○	—	—	—	—	—
(S)	○	○	—	—	—	—	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

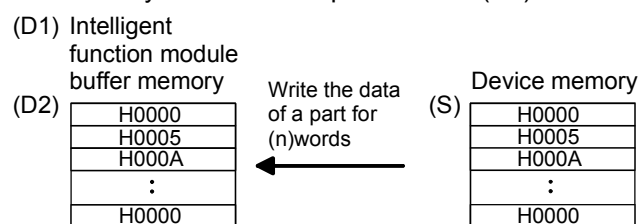
○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D1)	First I/O No. of the intelligent function module (000H to FF0H)	—
(D2)	First address of the buffer memory which writes data.	
(S)	Start device No. which writing data are stored.	
(n)	Number of words to be written (1 to 256)	

[Functions]

- (1) A part for (n)words of data from device specified with (S) are written to since address specified with (D2) of the buffer memory in the intelligent function module controlled by the self CPU specified with (D1).



- (2) First I/O No. of the module set by system setting is specified by (D1).

Power supply module	Q03UD CPU	Q173D CPU	QX40 First I/O No. No. : 00H	Q64AD First I/O No. No. : 10H	Q64DA First I/O No. No. : 20H	
---------------------	-----------	-----------	---------------------------------	----------------------------------	----------------------------------	--

(D1) sets 20H by the system setting when a TO instruction is executed in the D/A conversion module (Q64DA).

5 OPERATION CONTROL PROGRAMS

(3) The word devices that may be set at (D1), (D2), (S) and (n) are shown below.

Setting data	Word devices ^(Note-1)					Bit devices ^{(Note-1), (Note-2)}						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D1)	○	○	—	○	○	—	—	—	—	—	—	—
(D2)	○	○	—	○	○	—	—	—	—	—	—	—
(S)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(n)	○	○	—	○	○	—	—	—	—	—	—	—

(Note-1) : The device No. cannot be specified indirectly.

(Note-2) : Specify a multiple of 16 as the device number of bit data.

(Note-3) : PX and PY cannot be set.

(4) Adjust an executive task, the number of transfer word referring to the operation processing time so that this instruction may not obstruct the execution of the motion operation because processing time becomes long in argument to the number of words (n) to be written.

(5) The following analogue modules can be used as the Motion CPU control module.

- Q62AD-DGH
- Q62DA
- Q64AD
- Q62DA-FG
- Q64AD-GH
- Q64DA
- Q68ADV
- Q68DAV
- Q68ADI
- Q68DAI

[Errors]

(1) An operation error will occur if:

- Number of words (n) to be written is outside the range of 1 to 256.
- Motion CPU cannot communicate with intelligent function module at the instruction execution.
- Abnormalities of the intelligent function module were detected at the instruction execution.
- I/O No.s specified with (D1) differ from the intelligent function module controlled by the self CPU.
- The address specified with (D2) is outside the buffer memory range.
- Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range.
- (S) is a bit device and device number is not a multiple of 16.
- PX/PY is set in (S) to (S)+(n-1).

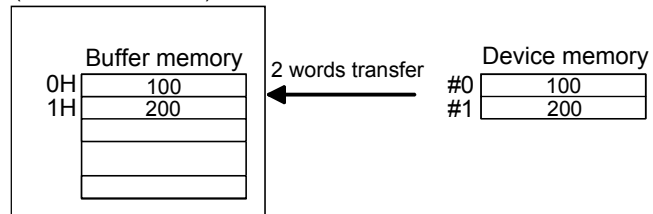
5 OPERATION CONTROL PROGRAMS

[Program examples]

- (1) 2 words from #0 is written to since buffer memory address of the intelligent function module (First I/O No. : 010H).

TO H010, H0, #0, K2

Intelligent function module
(First I/O No. : 010H)



5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.13.9 Read device data from intelligent function module : FROM

Format	FROM(D), (S1), (S2), (n)	Number of basic steps	7
--------	--------------------------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	○	—	—	—	—	—	—	—	—	—
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

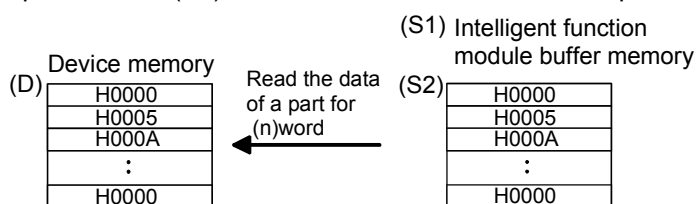
○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Start device No. which stores the reading data.	—
(S1)	First I/O No. of the intelligent function module (000H to FF0H)	
(S2)	First address of the buffer memory which it will be read.	
(n)	Number of words to be read (1 to 256)	

[Functions]

- (1) A part for (n)words of data are read from the address specified with (S2) of the buffer memory in the intelligent function module controlled by the self CPU specified with (S1), and are stored since the device specified with (S2).



- (2) First I/O No. of the module set by system setting is specified by (D1).

Power supply module	Q03UD CPU	Q173D CPU	QX40 First I/O No. No. : 00H	Q64AD First I/O No. No. : 10H	Q64DA First I/O No. No. : 20H	
---------------------	-----------	-----------	------------------------------------	-------------------------------------	-------------------------------------	--

(D1) sets 20H by the system setting when a TO instruction is executed in the D/A conversion module (Q64DA).

5 OPERATION CONTROL PROGRAMS

(3) The word devices that may be set at (D), (S1), (S2) and (n) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D)	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/> (Note-3)	<input type="radio"/> (Note-3)
(S1)	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	—	—	—	—	—	—	—
(S2)	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	—	—	—	—	—	—	—
(n)	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>	<input type="radio"/>	—	—	—	—	—	—	—

(Note-1) : The device No. cannot be specified indirectly.

(Note-2) : Specify a multiple of 16 as the device number of bit data.

(Note-3) : PX and PY cannot be set.

(4) Adjust an executive task, the number of transfer word referring to the operation processing time so that this instruction may not obstruct the execution of the motion operation because processing time becomes long in argument to the Number of words (n) to be read.

(5) The following analogue modules can be used as the Motion CPU control module.

- Q62AD-DGH
- Q62DA
- Q64AD
- Q62DA-FG
- Q64AD-GH
- Q64DA
- Q68ADV
- Q68DAV
- Q68ADI
- Q68DAI

[Errors]

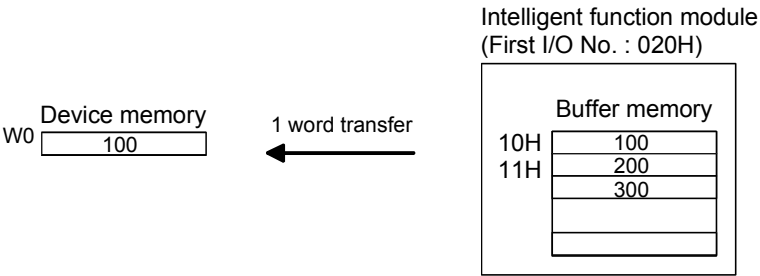
(1) An operation error will occur if:

- Number of words (n) to be read is outside the range of 1 to 256.
- Motion CPU cannot communicate with intelligent function module at the instruction execution.
- Abnormalities of the intelligent function module were detected at the instruction execution.
- I/O No.s specified with (S1) differ from the intelligent function module controlled by the self CPU.
- The address specified with (S2) is outside the buffer memory range.
- Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range.
- (D) is a bit device and device number is not a multiple of 16.
- PX/PY is set in (D) to (D)+(n-1).

[Program examples]

- (1) 1 word is read from the buffer memory address 10H of the intelligent function module (First I/O No. : 020H), and is stored in W0.

FROM W0, H020, H10, K1



5 OPERATION CONTROL PROGRAMS

F/FS	G
—	○

5.13.10 Time to wait : TIME

Format	TIME(S)	Number of basic steps	7
--------	---------	-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	—	—	○	○	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Waiting time (0 to 2147483647)[ms]	Logical type (true/false)

[Functions]

- (1) A wait state continues for the time specified with (S). The result is false when the elapsed time is less than the preset time, or the result is true and execution transits when the preset time has elapsed.
- (2) When a 16-bit integer type word device is used to specify any of 32768 to 65535[ms] at (S), convert it into an unsigned 32-bit integer value with ULONG. (Refer to the program example.)

[Errors]

- (1) An operation error will occur if:
 - (S) is an indirectly specified device and its device No. is outside the range. ; or
 - The data (device data at indirect specification) specified with (S) is outside the range of 0 to 2147483647.

[Program examples]

- (1) Program which sets a wait of 60 seconds (when constant is specified)

```
TIME K60000
```

- (2) Program for a case where there may be a wait of 32768 to 65535[ms] for 16-bit integer type indirect designation (#0)

```
TIME ULONG(#0)
```

- (3) Program which SETS (RSTs) a bit device when the specified time has elapsed

```
SET M100 = TIME K60000
```

POINT	
(1)	When the waiting time setting is indirectly specified with a word device, the value imported first is used as the device value for exercising control. The set time cannot be changed if the device value is changed during a wait state.
(2)	The TIME instruction is equivalent to a conditional expression, and therefore may be set on only the last line of a transition (G) program.
(3)	When the transition program (Gn) of the same number having the TIME instruction setting is used in multiple Motion SFC programs, avoid running them at the same time. (If they are run simultaneously, the waiting time in the program run first will be illegal.)
(4)	Another transition program (Gn) can executed a time of instruction by multiple Motion SFC program simultaneously. (Multi active step less than 256.)
(5)	While time by TIME instruction waits, the wait time can not be stopped.

5 OPERATION CONTROL PROGRAMS

F/FS	G
○	○

5.14 Comment Statement : //

Format	//
--------	----

Number of basic steps	—
-----------------------	---

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
—	—	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

There are no setting data.

[Functions]

- (1) A character string from after // to a block end is a comment.

[Errors]

- (1) There are no operation errors for comment: //.

[Program examples]

- (1) Example which has commented a substitution program.

D0=D1//Substitutes the D0 value (16-bit integer data) to D1.
--

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

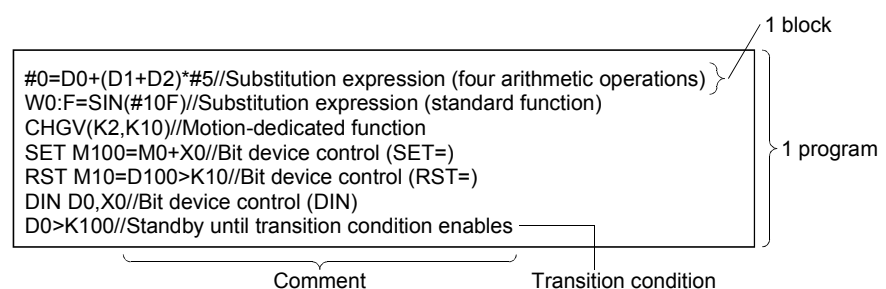
6. TRANSITION PROGRAMS

6.1 Transition Programs

(1) Transition programs

- (a) Substitution operation expressions, motion-dedicated functions, bit device control commands and transition conditions can be set in transition programs.
- (b) Multiple blocks can be set in one transition program.
- (c) There are no restrictions on the number of blocks that may be set in a single transition program.
Note that one program is within 64k bytes.
- (d) The maximum number of characters in one block is 128.
- (e) Transition condition must be set in the last block of a transition program.
Transition program is repeated until the transition condition enables, and when the transition condition has enabled, it shifts to the next step.
Transition condition can be set only in the last block.
- (f) As a special transition program, a program which only no operation (NOP) is set in one block can be created.
This program is used when it is not set as interlock to process to next step with completion of servo program.
Refer to Section "4.9 Branches, Couplings" for details.

A transition program example is shown below.



What can be set as a transition condition in the last block are bit conditional expressions, comparison conditional expressions and device set (SET=)/device reset (RST=) which return logical data values (true/false). In the case of device set (SET=)/device reset (RST=), whether the bit or comparison conditional expression specified at (S) is true or false is a transition condition, and when the transition condition enables, device set/reset is executed and execution shifts to the next step.

Transition condition description examples are given below.

Classification	Description example
Bit conditional expression	M0
	!M0+X10 * M100
Comparison conditional expression	(D0>K100)+(D100L!=K20L)
Device set (SET=)	SET Y0=M100
Device reset (RST=)	RST M10=D0==K100

POINT
<p>(1) A transition program differs from an operation control program in that a transition condition is set in the last block. Other settings are the same as those of the operation control program.</p> <p>(2) When setting device set (SET=)/device reset (RST=) in the last block as a transition condition, the bit or comparison conditional expression specified with (S) is not omissible.</p> <p>(3) Only the bit or comparison conditional expression cannot be set in other than the last block. Device set (SET=)/device reset (RST=) can be set in other than the last block.</p>

7. MOTION CONTROL PROGRAMS

7.1 Servo Instruction List

Table 7.1 lists servo instructions used in servo programs.

Refer to Section 7.2 to 7.4 for details of the current value change control (CHGA, CHGA-E, CHGA-C).

Refer to the "Q173DCPU/Q172DCPU Motion Controller (SV13/SV22) Programming Manual (REAL MODE)" for other servo instructions.

(1) Guide to servo instruction list

Table 7.1 Guide to Servo Instruction List

Positioning control	Instruction symbol	Processing	Positioning data																												Number of steps																
			Common								Arc/Helical			OSC			Parameter block										Other																				
			Parameter block No.	Axis	Address/travel	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Pitch	Starting angle	Amplitude	Frequency	Reference axis No.	Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing	Allowable error range for circular interpolation	S-curve ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel		Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration/deceleration time	Fixed position stop											
			Virtual enable																																												
			1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1														
			1	—	2	2	1	1	1	2	2	2	1	2	2	1	1	2	1	1	1	1	2	1	2	1	—	2	2	1	1	1	1														
1 axis	ABS-1	Absolute 1-axis positioning	△	○	○	○	△	△										△	△	△	△	△	△	△				△																			
	INC-1	Incremental 1-axis positioning	△	○	○	○	△	△											△	△	△	△	△	△				△																			
	ABS-2	Absolute 2-axes linear														○	△	△	△	△	△	△	△	△				△																			
axes																																		4 to 17													
1)																															2)																











Number	Description	
1)	Instruction symbol	Gives the servo instructions usable in servo programs.
	Processing	Gives the processing outlines of the servo instructions.
2)	(a) Indicates positioning data which can be set in servo instructions. 1) ○: Item which must be set (Data which cannot execute the servo instruction unless it sets.) 2) △: Item which is set when required (Data which will be controlled by the default value unless it sets.)	
	(b) Allows direct or indirect designation (except axis No.) 1) Direct designation : Set with numerical value. 2) Indirect designation : Set with word device. • Servo program execution is controlled using the preset word device contents. • Each setting item may either be 1 or 2 word data. • For 2 word data, set the start device No..	
	(c) Number of steps As there are more setting items, there are more number of instruction steps. (The number of steps is displayed when a servo program is created.) (The instruction + ○ item comprise the minimum steps, and one △ item increases the number of steps by 1.)	
3)	Items common to the servo instructions	
4)	Items set in circular interpolation starting servo programs	
5)	Items set for high-speed oscillation	
6)	Set when changing the parameter block (default value when not set) data set in the servo program to control. (The parameter block data are not changed.)	
7)	Setting items other than the common, arc and parameter block items (Items to be set vary with the servo instruction.)	
8)	Indicates the number of steps of each servo instruction.	

7 MOTION CONTROL PROGRAMS

(2) Servo instruction list

Table 7.2 indicates the servo instructions available for servo programs and the positioning data set in servo instructions.

Table 7.2 Servo Instruction List

Positioning control		Instruction symbol	Processing	Positioning data										
				Common							Arc/Helical			
				Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Pitch
				Virtual enable	○	○	○	○	○	—	○	○	○	○
			Number of steps	1	1	1	1	1	1	1	1	1		
			Number of indirect words	1	—	2	2	1	1	1	2	2	1	
Linear interpolation control	1 axis	ABS-1	Absolute 1-axis positioning	△	○	○	○	△	△					
		INC-1	Incremental 1-axis positioning	△	○	○	○	△	△					
	2 axes	ABS-2	Absolute 2-axes linear interpolation	△	○	○	○	△	△					
		INC-2	Incremental 2-sxes linear interpolation	△	○	○	○	△	△					
	3 axes	ABS-3	Absolute 3-axes linear interpolation	△	○	○	○	△	△					
		INC-3	Incremental 3-axes linear interpolation	△	○	○	○	△	△					
	4 axes	ABS-4	Absolute 4-axes linear interpolation	△	○	○	○	△	△					
		INC-4	Incremental 4-axes linear interpolation	△	○	○	○	△	△					
Circular interpolation control	Auxiliary point-specified	ABS 	Absolute auxiliary point-specified circular interpolation	△	○	○	○	△	△		○			
		INC 	Incremental auxiliary point-specified circular interpolation	△	○	○	○	△	△		○			
	Radius-specified	ABS 	Absolute radius-specified circular interpolation less than CW 180°	△	○	○	○	△	△			○		
		ABS 	Absolute radius-specified circular interpolation CW 180° or more	△	○	○	○	△	△			○		
		ABS 	Absolute radius-specified circular interpolation less than CCW 180°	△	○	○	○	△	△			○		
		ABS 	Absolute radius-specified circular interpolation CCW 180° or more	△	○	○	○	△	△			○		
		INC 	Incremental radius-specified circular interpolation less than CW 180°	△	○	○	○	△	△			○		
		INC 	Incremental radius-specified circular interpolation CW 180° or more	△	○	○	○	△	△			○		
		INC 	Incremental radius-specified circular interpolation less than CCW 180°	△	○	○	○	△	△			○		
		INC 	Incremental radius-specified circular interpolation CCW 180° or more	△	○	○	○	△	△			○		



















7 MOTION CONTROL PROGRAMS

Positioning data																						Number of steps
OSC			*1 Reference axis No.	Parameter block									Others									
Starting angle	Amplitude	Frequency		Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time	Fixed position stop	
—	—	—		○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	—	
1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
2	2	2	1	1	2	1	1	1	1	1	2	1	*2 1/ 1(B)	—	2	*2 1(B)	*2 1(B)	1	*2 1(B)	1	*2 1(B)	
					△	△	△	△	△	△		△				△					4 to 17	
					△	△	△	△	△	△		△				△						
			○	△	△	△	△	△	△	△		△				△					5 to 20	
			○	△	△	△	△	△	△	△		△				△						
			○	△	△	△	△	△	△	△		△				△					7 to 21	
			○	△	△	△	△	△	△	△		△				△						
			○	△	△	△	△	△	△	△		△				△					8 to 22	
			○	△	△	△	△	△	△	△		△				△						
				△	△	△	△	△	△	△	△	△				△					7 to 22	
				△	△	△	△	△	△	△	△	△				△						
				△	△	△	△	△	△	△	△	△				△					6 to 21	
				△	△	△	△	△	△	△	△	△				△						
				△	△	△	△	△	△	△	△	△				△						
				△	△	△	△	△	△	△	△	△				△						
				△	△	△	△	△	△	△	△	△				△						
				△	△	△	△	△	△	△	△	△				△						
				△	△	△	△	△	△	△	△	△				△						
				△	△	△	△	△	△	△	△	△				△						
				△	△	△	△	△	△	△	△	△				△						

○ : Must be set. △ : Set if required.
 *1 : Only reference axis speed specification.
 *2 : (B) indicates a bit device.

7 MOTION CONTROL PROGRAMS

Table 7.2 Servo Instruction List (continued)

Positioning control		Instruction symbol	Processing	Positioning data										
				Common							Arc/Helical			
				Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Pitch
				Virtual enable	○	○	○	○	○	○	—	○	○	○
		Number of steps	1	1	1	1	1	1	1	1	1	1	1	
		Number of indirect words	1	—	2	2	1	1	1	2	2	2	1	
Circular interpolation control	Central point-specified	ABS 	Absolute central point-specified circular interpolation CW	△	○	○	○	△	△				○	
		ABS 	Absolute central point-specified circular interpolation CCW	△	○	○	○	△	△				○	
		INC 	Incremental central point-specified circular interpolation CW	△	○	○	○	△	△				○	
		INC 	Incremental central point-specified circular interpolation CCW	△	○	○	○	△	△				○	
Helical interpolation control	Auxiliary point-specified	ABH 	Absolute auxiliary point- specified helical interpolation	△	○	○	○	△	△		○			○
		INH 	Incremental auxiliary point- specified helical interpolation	△	○	○	○	△	△		○			○
	Radius-specified	ABH 	Absolute radius-specified helical interpolation less than CW 180°	△	○	○	○	△	△			○		○
		ABH 	Absolute radius-specified helical interpolation CW 180° or more	△	○	○	○	△	△			○		○
		ABH 	Absolute radius-specified helical interpolation less than CCW 180°	△	○	○	○	△	△			○		○
		ABH 	Absolute radius-specified helical interpolation CCW 180° or more	△	○	○	○	△	△			○		○
		INH 	Incremental radius-specified helical interpolation less than CW 180°	△	○	○	○	△	△			○		○
		INH 	Incremental radius-specified helical interpolation CW 180° or more	△	○	○	○	△	△			○		○
		INH 	Incremental radius-specified helical interpolation less than CCW 180°	△	○	○	○	△	△			○		○
		INH 	Incremental radius-specified helical interpolation CCW 180° or more	△	○	○	○	△	△			○		○
	Central point-specified	ABH 	Absolute central point-specified helical interpolation CW	△	○	○	○	△	△				○	○
		ABH 	Absolute central point-specified helical interpolation CCW	△	○	○	○	△	△				○	○
		INH 	Incremental central point-specified helical interpolation CW	△	○	○	○	△	△				○	○
		INH 	Incremental central point-specified helical interpolation CCW	△	○	○	○	△	△				○	○

7 MOTION CONTROL PROGRAMS

	Positioning data																				Number of steps		
	OSC			*1 Reference axis No.	Parameter block								Others										
	Starting angle	Amplitude	Frequency		Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF		Fixed position stop acceleration /deceleration time	Fixed position stop
—	—	—	○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	○	—	—	
1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
2	2	2	2	1	1	2	1	1	1	1	1	2	1	*2 1/ 1(B)	—	2	*2 1(B)	*2 1(B)	1	*2 1(B)	1	*2 1(B)	
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						
					△	△	△	△	△	△	△	△	△				△						

○ : Must be set. △ : Set if required.

*1 : Only reference axis speed specification.

*2 : (B) indicates a bit device.

7 MOTION CONTROL PROGRAMS

Table 7.2 Servo Instruction List (continued)

Positioning control	Instruction symbol	Processing	Positioning data										
			Common							Arc/Helical			
			Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Pitch
			Virtual enable	○	○	○	○	○	—	○	○	○	○
			Number of steps	1	1	1	1	1	1	1	1	1	1
			Number of indirect words	1	—	2	2	1	1	2	2	2	1
Fixed-pitch feed	1 axis	FEED-1	1-axis fixed-pitch feed start	△	○	○	○	△	△				
	2 axes	FEED-2	2-axes linear interpolation fixed-pitch feed start	△	○	○	○	△	△				
	3 axes	FEED-3	3-axes linear interpolation fixed-pitch feed start	△	○	○	○	△	△				
Speed control (I)	Forward rotation	VF	Speed control (I) forward rotation start	△	○		○		△				
	Reverse rotation	VR	Speed control (I) reverse rotation start	△	○		○		△				
Speed control (II)	Forward rotation	VVF	Speed control (II) forward rotation start	△	○		○		△	△			
	Reverse rotation	VVR	Speed control (II) reverse rotation start	△	○		○		△	△			
Speed-position control	Forward rotation	VPF	Speed-position control forward rotation start	△	○	○	○	△	△	△			
	Reverse rotation	VPR	Speed-position control reverse rotation start	△	○	○	○	△	△	△			
	Restart	VPSTART	Speed-position control restart		○								
Speed-switching control		VSTART	Speed-switching control start	△									
		VEND	Speed-switching control end										
		ABS-1	Speed-switching control end point address		○	○	○	△	△	△			
		ABS-2			○	○	○	△	△	△			
		ABS-3			○	○	○	△	△	△			
		INC-1	Travel value up to speed-switching control end point		○	○	○	△	△	△			
		INC-2			○	○	○	△	△	△			
		INC-3			○	○	○	△	△	△			
		VABS	Speed-switching point absolute specification			○	○		△	△			
		VINC	Speed-switching point incremental specification			○	○		△	△			

7 MOTION CONTROL PROGRAMS

Positioning data																						Number of steps
OSC			*1 Reference axis No.	Parameter block									Others									
Starting angle	Amplitude	Frequency		Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time	Fixed position stop	
—	—	—	○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	—	—	
1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
2	2	2	1	1	2	1	1	1	1	1	2	1	*2 1/ 1(B)	—	2	*2 1(B)	*2 1(B)	1	*2 1(B)	1	*2 1(B)	
					△	△	△	△	△	△		△				△					4 to 17	
				△	△	△	△	△	△	△		△				△					5 to 19	
				△	△	△	△	△	△	△		△				△					7 to 21	
					△	△	△	△	△	△		△				△					3 to 15	
					△	△	△	△	△	△		△				△						
					△	△	△	△	△	△		△				△					3 to 16	
					△	△	△	△	△	△		△				△						
					△	△	△	△	△	△		△				△					4 to 18	
					△	△	△	△	△	△		△				△						
																△					2 to 4	
				△	△	△	△	△	△	△		△				△					1 to 13	
																					1	
																△					4 to 9	
																△					5 to 10	
																△					7 to 12	
																△					4 to 9	
																△					5 to 10	
																△					7 to 12	
																					4 to 6	

○ : Must be set. △ : Set if required.

*1 : Only reference axis speed specification.

*2 : (B) indicates a bit device.

7 MOTION CONTROL PROGRAMS

Table 7.2 Servo Instruction List (continued)

Positioning control			Instruction symbol	Processing	Positioning data										
					Common							Arc/Helical			
					Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Pitch
				Virtual enable	○	○	○	○	○	—	○	○	○	○	
				Number of steps	1	1	1	1	1	1	1	1	1	1	
				Number of indirect words	1	—	2	2	1	1	1	2	2	2	1
Speed control with fixed position stop	Forward rotation	<div>PVF</div>	Speed control with fixed position stop absolute specification	△	○	○	○	△	△						
	Reverse rotation	<div>PVR</div>		△	○	○	○	△	△						
Position follow-up control		<div>PFSTART</div>	Position follow-up control start	△	○	○	○		△						
Constant-speed control		<div>CPSTART1</div>	1-axis constant-speed control start	△	○		○								
		<div>CPSTART2</div>	2-axes constant-speed control start	△	○		○								
		<div>CPSTART3</div>	3-axes constant-speed control start	△	○		○								
		<div>CPSTART4</div>	4-axes constant-speed control start	△	○		○								
		<div>ABS-1</div>	Constant-speed control passing point absolute specification		○	○			△	△					
		<div>ABS-2</div>			○	○			△	△					
		<div>ABS-3</div>			○	○			△	△					
		<div>ABS-4</div>			○	○			△	△					
		<div>ABS</div>			○	○			△	△	○				
		<div>ABS</div>			○	○			△	△		○			
		<div>ABS</div>			○	○			△	△		○			
		<div>ABS</div>			○	○			△	△		○			
		<div>ABS</div>			○	○			△	△		○			
		<div>ABS</div>			○	○			△	△			○		
		<div>ABS</div>			○	○			△	△			○		
		<div>ABH</div>	Constant-speed control passing point helical absolute specification		○	○			△	△	○			○	
		<div>ABH</div>			○	○			△	△		○		○	
		<div>ABH</div>			○	○			△	△		○		○	
		<div>ABH</div>			○	○			△	△		○		○	
		<div>ABH</div>			○	○			△	△		○		○	
		<div>ABH</div>			○	○			△	△			○	○	
		<div>ABH</div>			○	○			△	△			○	○	
		<div>ABH</div>			○	○			△	△			○	○	

7 MOTION CONTROL PROGRAMS

Positioning data																						Number of steps
OSC			*1 Reference axis No.	Parameter block									Others									
Starting angle	Amplitude	Frequency		Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time	Fixed position stop	
—	—	—		○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	—	
1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
2	2	2	1	1	2	1	1	1	1	1	2	1	*2 1/ 1(B)	—	2	*2 1(B)	*2 1(B)	1	*2 1(B)	1	*2 1(B)	
					△		△	△	△	△		△				△				○	○	6 to 19
					△		△	△	△	△		△				△				○	○	
					△	△	△	△	△	△		△				△						4 to 16
					△	△	△	△	△	△		△				△		△				3 to 15
				△	△	△	△	△	△	△	△	△				△		△				3 to 17
				△	△	△	△	△	△	△	△	△				△		△				4 to 17
				△	△	△	△	△	△	△	△	△				△		△				
															△		△		△			2 to 10
															△		△		△			3 to 11
															△		△		△			4 to 12
															△		△		△			5 to 13
															△		△		△			5 to 14
															△		△		△			4 to 13
															△		△		△			
															△		△		△			
															△		△		△			5 to 14
															△		△		△			9 to 14
															△		△		△			8 to 13
															△		△		△			
															△		△		△			
															△		△		△			9 to 14














○ : Must be set. △ : Set if required.

*1 : Only reference axis speed specification.

*2 : (B) indicates a bit device.

7 MOTION CONTROL PROGRAMS

Table 7.2 Servo Instruction List (continued)

Positioning control	Instruction symbol	Processing	Positioning data										
			Common							Arc/Helical			
			Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Pitch
			Virtual enable	○	○	○	○	○	—	○	○	○	○
		Number of steps	1	1	1	1	1	1	1	1	1		
		Number of indirect words	1	—	2	2	1	1	1	2	2	2	1
Constant-speed control	INC-1	Constant-speed control passing point incremental specification		○	○			△	△				
	INC-2			○	○			△	△				
	INC-3			○	○			△	△				
	INC-4			○	○			△	△				
	INC 			○	○			△	△	○			
	INC 			○	○			△	△		○		
	INC 			○	○			△	△		○		
	INC 			○	○			△	△		○		
	INC 			○	○			△	△			○	
	INC 			○	○			△	△			○	
	INH 		Constant-speed control passing point helical incremental specification		○	○			△	△	○		
	INH 			○	○			△	△		○		○
	INH 			○	○			△	△		○		○
	INH 			○	○			△	△		○		○
	INH 			○	○			△	△			○	○
	INH 			○	○			△	△			○	○
	INH 			○	○			△	△			○	○
	CPEND	Constant-speed control end					△						

7 MOTION CONTROL PROGRAMS

Positioning data																					Number of steps		
OSC			*1 Reference axis No.	Parameter block									Others										
Starting angle	Amplitude	Frequency		Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time		Fixed position stop	
—	—	—		○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○		—	—
1	1	1		1	1	2	1	1	1	1	1	1	1	1	1	2	2	2	1	2		1	1
2	2	2	1	1	2	1	1	1	1	1	2	1	*2 1/ 1(B)	—	2	*2 1(B)	*2 1(B)	1	*2 1(B)	1	*2 1(B)		
															△		△		△		2 to 10		
															△		△		△		3 to 11		
															△		△		△		4 to 12		
															△		△		△		5 to 13		
															△		△		△		5 to 14		
															△		△		△		4 to 13		
															△		△		△				
															△		△		△				
															△		△		△				
															△		△		△		5 to 14		
															△		△		△		9 to 14		
															△		△		△		8 to 13		
															△		△		△				
															△		△		△				
															△		△		△				
															△		△		△		9 to 14		
															△		△		△		1 to 2		

○ : Must be set. △ : Set if required.

*1 : Only reference axis speed specification.

*2 : (B) indicates a bit device.

Table 7.2 Servo Instruction List (continued)

Positioning control	Instruction symbol	Processing	Positioning data										
			Common							Arc/Helical			
			Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Pitch
		Virtual enable	○	○	○	○	○	○	—	○	○	○	○
		Number of steps	1	1	1	1	1	1	1	1	1	1	1
		Number of indirect words	1	—	2	2	1	1	1	2	2	2	1
Repetition of same control (used in speed control, constant-speed control)	FOR-TIMES	Repeat range start setting											
	FOR-ON												
	FOR-OFF												
	NEXT	Repeat range end setting											
Simultaneous start	START	Simultaneous start											
Home position return	ZERO	Home position return start		○									
High speed oscillation	OSC	High-speed oscillation	△	○				△					
Current Value change	CHGA	Servomotor/Virtual Servomotor Shaft Current Value Change		○	○								
	CHGA-E	Encoder current value change		○	○								
	CHGA-C	CAM shaft current value change		○	○								

7 MOTION CONTROL PROGRAMS

	Positioning data																				Number of steps		
	OSC			*1 Reference axis No.	Parameter block								Others										
	Starting angle	Amplitude	Frequency		Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF		Fixed position stop acceleration /deceleration time	Fixed position stop
	—	—	—		○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○		○	—
	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
	2	2	2	1	1	2	1	1	1	1	1	2	1	*2 1/ 1(B)	—	2	*2 1(B)	*2 1(B)	1	*2 1(B)	1	*2 1(B)	
														○								2	
														○									
														○									
																						3	
															○							2 to 3	
																						2	
	○	○	○							△							△					5 to 10	
																						3	

○ : Must be set. △ : Set if required.

*1 : Only reference axis speed specification.

*2 : (B) indicates a bit device.

7 MOTION CONTROL PROGRAMS

7.2 Servomotor/Virtual Servomotor Shaft Current Value Change

The current value of the specified axis is changed in the real mode.

The current value of the specified virtual servomotor shaft is changed in the virtual mode.

Servo instruction	Positioning method	Number of Control axes	Items set using MT Developer																		Speed change
			Common						Arc		Parameter block								Others		
			Parameter block No.	Axis	Address/travel	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value Deceleration processing at stop input Allowable error range for circular interpolation	S-curve ratio	Cancel	
CHGA	Absolute	1		○	○															Disable	

○ : Item which must be set

△ : Item which is set when required

[Controls]

Control using CHGA instruction

- (1) Executing the CHGA instruction changes the current value in the following procedure.
 - (a) The start accept flag (M2001 to M2008/M2001 to M2032) corresponding to the specified axis is turned on.
 - (b) The current value of the specified axis is changed to the specified address.
 - (c) Start accept flag is turned off at completion of the current value change.
- (2) The current value of the specified axis is changed in the real mode.
- (3) The current value of the specified virtual servo-motor shaft is changed in the virtual mode.
- (4) The used axis No. can be set within the following range.

Q172DCPU	Q173DCPU
Axis 1 to 8	Axis 1 to 32

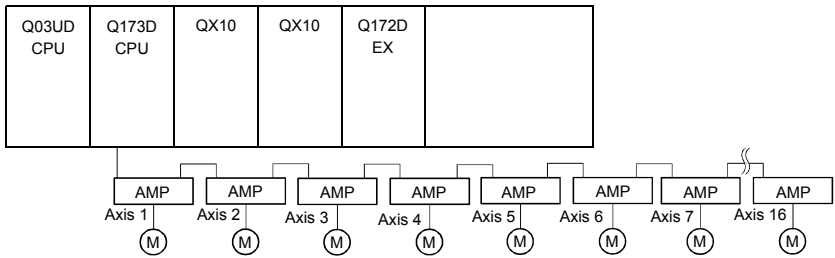
- (5) The address which made the current value change by CHGA instruction is valid on the power supply turning on.

[Program example]

A program which made the current value change control in the real mode is described as the following conditions.

(1) System configuration

The current value change control of axis 2 is executed.

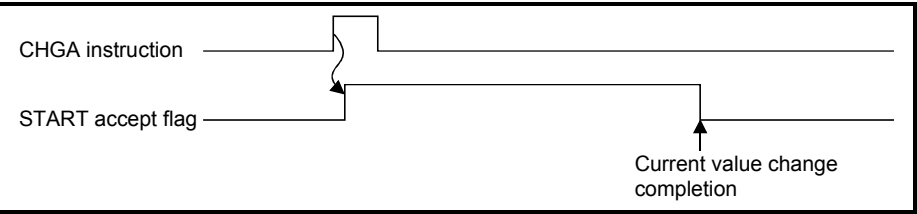


(2) The current value change control conditions

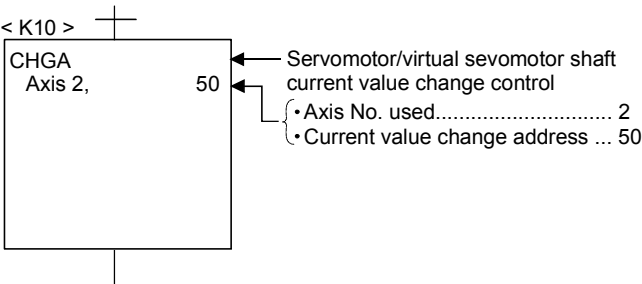
(a) The current value change control conditions are shown below.

Item	Setting
Servo program No.	10
Control axis No.	2
Current value change address	50

(3) Operation timing



(4) Servo program



POINT

Current value changing instructions

- When PLC ready flag (M2000) or PCPU ready flag (SM500) is OFF, a minor error ^(Note) [100] occurs and a current value change is not made.
- This change is made only during a stop. If a current value change is made while the specified axis is starting, a minor error ^(Note) [101] (start accept signal of the corresponding axis is ON) occurs and the current value change is not made.
- If the servo of the corresponding axis is not READY, a major error ^(Note) [1004] occurs and the current value change is not made.
- If the corresponding axis is in a servo error, a major error ^(Note) [1005] occurs and the current value change is not made.

For SV22

- The current value change of specified axis is executed in the real mode, and the current value change of specified servomotor axis is executed in the virtual mode.
- Set the current value change program of the virtual servomotor shaft within the virtual mode program No. range set in "program mode assignment".
- Set the current value change program of the servomotor (output) shaft within the real mode program No. range.
- If a virtual servomotor shaft current value change is executed in the real mode, a servo program setting error ^(Note) [903] occurs and the current value change is not made.
- If a servomotor (output) shaft current value change is executed in the virtual mode, a servo program setting error ^(Note) [904] occurs and the current value change is not made.
- If a current value change is made during mode changing, a servo program setting error ^(Note) [907] (real mode → virtual mode changing) or [908] (virtual mode → real mode changing) occurs and the current value change is not made.

(Note) : Refer to the "Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)"/ "Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for minor error, major error and servo program setting error.

7 MOTION CONTROL PROGRAMS

7.3 Synchronous Encoder Shaft Current Value Change Control (SV22 only)

The current value of the specified synchronous encoder shaft is changed in the virtual mode.

Servo instruction	Positioning method	Number of Control axes	Items set using MT Developer																			Speed change	
			Common						Arc		Parameter block									Others			
			Parameter block No.	Axis	Address/travel	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio		Cancel
CHGA-E	Absolute	1		<input type="radio"/>	<input type="radio"/>																		Disable

○ : Item which must be set

△ : Item which is set when required

[Controls]

Control using CHGA-E instruction

- (1) Executing the CHGA-E instruction changes the current value of the synchronous encoder shaft in the following procedure.
 - (a) The synchronous encoder shaft current value changing flag (M2101 to M2112) corresponding to the specified synchronous encoder shaft is turned on.
 - (b) The current value of the specified synchronous encoder shaft is changed to the specified address.
 - (c) The synchronous encoder shaft current value changing flag is turned off at completion of the current value change.
- (2) The used axis No. can be set within the following range.

Q172DCPU	Q173DCPU
Axis 1 to 8	Axis 1 to 12

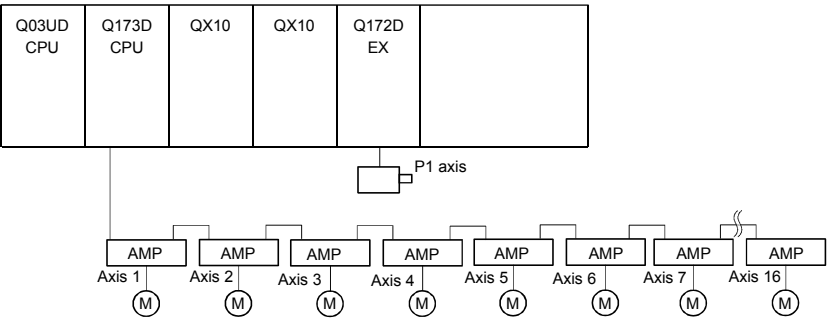
- (3) The address which made the current value change by CHGA-E instruction is valid after also the power supply turned off.

[Program example]

A program which made the current value change control of the synchronous encoder shaft is described as the following conditions.

(1) System configuration

The current value change control of the synchronous encoder shaft P1 is executed.

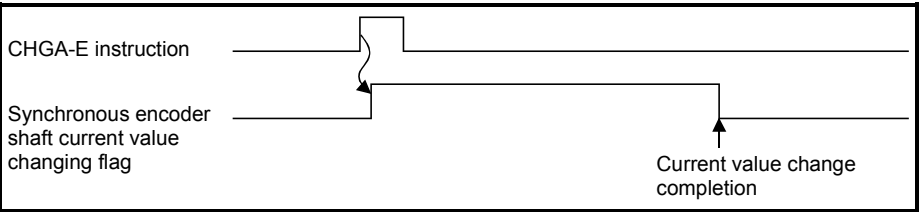


(2) The current value change control conditions

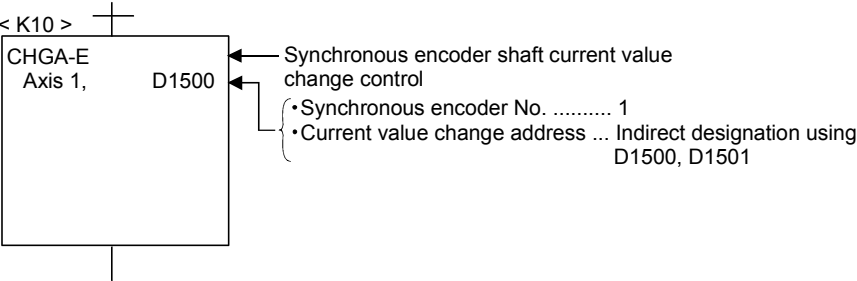
(a) The current value change control conditions are shown below.

Item	Setting
Servo program No.	10
Synchronous encoder No.	1
Current value change address	Indirect designation using D1500, D11501

(3) Operation timing



(4) Servo program



POINT	
	<p><u>Synchronous encoder current value changing instructions</u></p> <ul style="list-style-type: none"> • The current value change of the synchronous encoder is executed if operation is being performed in the virtual mode (during pulse input from the synchronous encoder). If the current value is changed, the feed current value of the synchronous encoder continues from the new value. • The current value change of the synchronous encoder does not affect the current value of the output module. • Set the current value change program of the synchronous encoder shaft program within the virtual mode program No. range set in "program mode assignment". • When PLC ready flag (M2000) or PCPU ready flag (SM500) is OFF, a minor error ^(Note) [100] occurs and a current value change is not made. • If a synchronous encoder current value change is executed in the real mode, a servo program setting error ^(Note) [903] or [905] occurs and the current value change is not made. ([903] when the current value change servo program is set to within the virtual mode program No. range, or 905 when it is set to within the real mode program No. range.) • If a current value change is made during mode changing, a servo program setting error ^(Note) [907] (real mode → virtual mode changing) or [908] (virtual mode → real mode changing) occurs and the current value change is not made.

(Note) : Refer to the "Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)"/"Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for minor error, major error and servo program setting error.

7 MOTION CONTROL PROGRAMS

7.4 Cam Shaft Within-One-Revolution Current Value Change Control (SV22 only)

The current value of the specified cam shaft within-one-revolution is changed in the virtual mode.

Servo instruction	Positioning method	Number of Control axes	Items set using MT Developer																Speed change		
			Common						Arc		Parameter block								Others		
			Parameter block No.	Axis	Address/travel	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value Deceleration processing at stop input Allowable error range for circular interpolation	S-curve ratio	Cancel	FIN acceleration/deceleration
CHGA-C	Absolute	1		<input type="radio"/>	<input type="radio"/>																Disable

○ : Item which must be set

△ : Item which is set when required

[Controls]

Control using CHGA-C instruction

- (1) Executing the CHGA-C instruction changes the within-one-revolution current value of the specified cam shaft to the address.
- (2) The cam shaft may be starting.
- (3) The used axis No. can be set within the following range.

Q172DCPU	Q173DCPU
Axis 1 to 8	Axis 1 to 32

- (4) The address which made the current value change by the CHGA-C instruction is valid after also the power supply turned off.

[Program example]

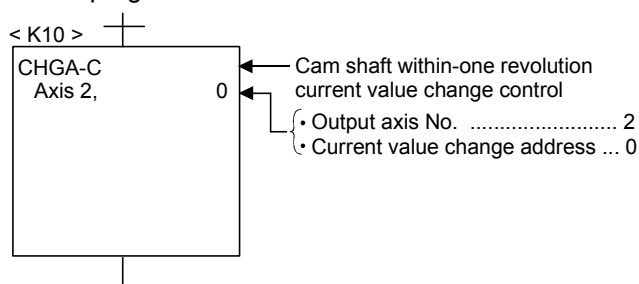
A program which made the current value change control of the cam shaft within-one-revolution current value change is described as the following conditions.

(1) Current value change control conditions

(a) The current value change control conditions are shown below.

Item	Setting
Servo program No.	10
Output axis No.	2
Current value change address	0

(2) Servo program



POINT

Cam shaft within-one revolution current value changing instructions

- If a new within-one revolution current value is outside the range 0 to (one-revolution pulse count - 1), a minor error ^(Note) [6120] occurs and current value change is not.
- Set the current value change program the cam shaft within-one-revolution within the virtual mode program No. range set in "program mode assignment".
- When PLC ready flag (M2000) or PCPU ready flag (SM500) is OFF, a minor error ^(Note) [100] occurs and a current value change is not made.
- If the cam shaft within-one-revolution current value change is executed in the real mode, a servo program setting error ^(Note) [903] or [905] occurs and the current value change is not made. ([903] when the current value change servo program is set to within the virtual mode program No. range, or 905 when it is set to within the real mode program No. range.)
- If a current value change is made during mode changing, a servo program setting error ^(Note) [907] (real mode → virtual mode changing) or [908] (virtual mode → real mode changing) occurs and the current value change is not made.

(Note) : Refer to the "Q173DCPU/Q172DCPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)"/"Q173DCPU/Q172DCPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for minor error, major error and servo program setting error.

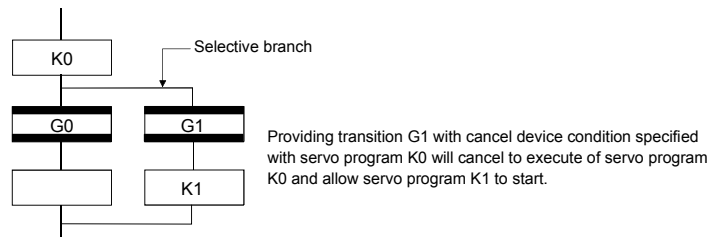
7 MOTION CONTROL PROGRAMS

7.5 Programming Instructions

7.5.1 Cancel • start

When a cancel start has been set in the setting items of the servo program which was started at the motion control step of the Motion SFC program, the cancel of the running servo program is valid but the servo program specified to start after a cancel is ignored, without being started.

The following example shows the Motion SFC program which exercises control equivalent to a cancel start.



7.5.2 Indirect designation using motion devices

The coasting timer (FT) cannot be used to make indirect specification in the servo program and mechanical system program.

8. MOTION DEVICES

The motion registers (#0 to #8735) and coasting timer (FT) are available as Motion CPU-dedicated devices.

They can be used in operation control (F/FS) programs or transition (G) programs.

8.1 Motion Registers (#0 to #8735)

Motion device	Item	Specifications
Motion register (#)	Number of points	8736 points (#0 to #8735)
	Data size	16-bit/point
	Latch	Only a user device is latched. (All points are cleared by latch clear operation.)
	Usable tasks	Normal, event and NMI
	Access	Read and write enabled in whole range

(1) Motion register list

- Common to all operating system software

Device No.	Application	Signal direction
#0 to	User devices (8000 points)	• Cleared by latch clear.
#8000 to	Monitor devices (640 points)	• Cleared at power on or reset only.
#8640 to #8735	Motion error history devices (96 points)	• Cleared at power on or reset only. • Cleared by the Motion error history request flag on (keep at power on or reset).

8 MOTION DEVICES

(a) Monitor devices (#8000 to #8639)

Information for each axis is stored in the monitor devices.

The details of the storage data are shown below.

Axis No.	Device No.	Signal name			
1	#8000 to #8019				
2	#8020 to #8039				
3	#8040 to #8059				
4	#8060 to #8079				
5	#8080 to #8099				
6	#8100 to #8119				
7	#8120 to #8139				
8	#8140 to #8159				
9	#8160 to #8179				
10	#8180 to #8199				
11	#8200 to #8219				
12	#8220 to #8239				
13	#8240 to #8259				
14	#8260 to #8279				
15	#8280 to #8299				
16	#8300 to #8319				
17	#8320 to #8339				
18	#8340 to #8359				
19	#8360 to #8379				
20	#8380 to #8399				
21	#8400 to #8419				
22	#8420 to #8439				
23	#8440 to #8459				
24	#8460 to #8479				
25	#8480 to #8499				
26	#8500 to #8519				
27	#8520 to #8539				
28	#8540 to #8559				
29	#8560 to #8579				
30	#8580 to #8599				
31	#8600 to #8619				
32	#8620 to #8639				

	Signal name	Refresh cycle	Signal direction
0	Servo amplifier type	When the servo amplifier power-on	Monitor device
1	Motor current	Operation cycle 1.7[ms] or less: Operation cycle Operation cycle 3.5[ms] or more: 3.5[ms]	
2	Motor speed		
3			
4	Command speed	Operation cycle	
5			
6	Home position return re-travel	At home position return re-travel	
7	value (Real mode only)		
8	Unusable	—	—
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			

8 MOTION DEVICES

(b) Motion error history devices (#8640 to #8735)

The Motion error history devices are shown below.

Device No.	Signal name	Signal direction		Refresh cycle	Fetch cycle
		Status	Command		
#8640 to #8651	Seventh error information in past (Oldest error information)	○	—	At error occurrence	—
#8652 to #8663	Sixth error information in past				
#8664 to #8675	Fifth error information in past				
#8676 to #8687	Fourth error information in past				
#8688 to #8699	Third error information in past				
#8700 to #8711	Second error information in past				
#8712 to #8723	First error information in past				
#8724 to #8735	Latest error information				

(Note-1) : Refer to Section "12.2 Motion Error Related Device" for the Motion error history.

8.2 Coasting Timer (FT)

Motion device	Item	Specification
Coasting timer (FT)	Number of points	1 point (FT)
	Data size	32-bit/point (-2147483648 to 2147483647)
	Latch	No latch. Cleared to zero at power-on or reset, a count rise is continued from now on.
	Usable tasks	Normal, event, NMI
	Access	Read only enabled
	Timer specifications	888 μ s timer (Current value (FT) is incremented by 1 per 888 μ s.)

9. OPERATION FOR MOTION SFC AND PARAMETER

9.1 Task Definitions

When to execute the Motion SFC program processing can be set only once in the program parameter per program.

Roughly classified, there are the following three different tasks.

Task type	Contents
Normal task	Executes in Motion CPU main cycle (free time).
Event task	1. Executes in fixed cycle (0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms). 2. Executes when the input set to the event task factor among external interrupts (16 points of QI60) turns on. 3. Executes by an interrupt from the PLC CPU.
NMI task	Executes when the input set to the NMI task factor among external interrupts (16 points of QI60) turns on.

9.2 Number of Consecutive Transitions and Task Operation

9.2.1 Number of consecutive transitions

With "execution of active step → judgment of next transition condition → transition processing performed when condition enables (transition of active step)" defined as a single basic operation of the Motion SFC program execution control in the execution cycle of the corresponding task, this operation is performed for the number of active steps to terminate processing once. And the same operation is processed continuously in the next cycle.

In this case, the transition destination step is executed in the next cycle when the transition condition enables.

Consecutive transition control indicates that transition destination steps are executed one after another in the same one execution cycle when their transition conditions have enabled (single basic operation is performed consecutively).

In this case, set the number of consecutive transitions.

Control exercised is common to the Motion SFC programs executed by normal tasks.

POINT	Set the number of consecutive transitions to the Motion SFC programs executed by event and NMI tasks for every program.
--------------	---

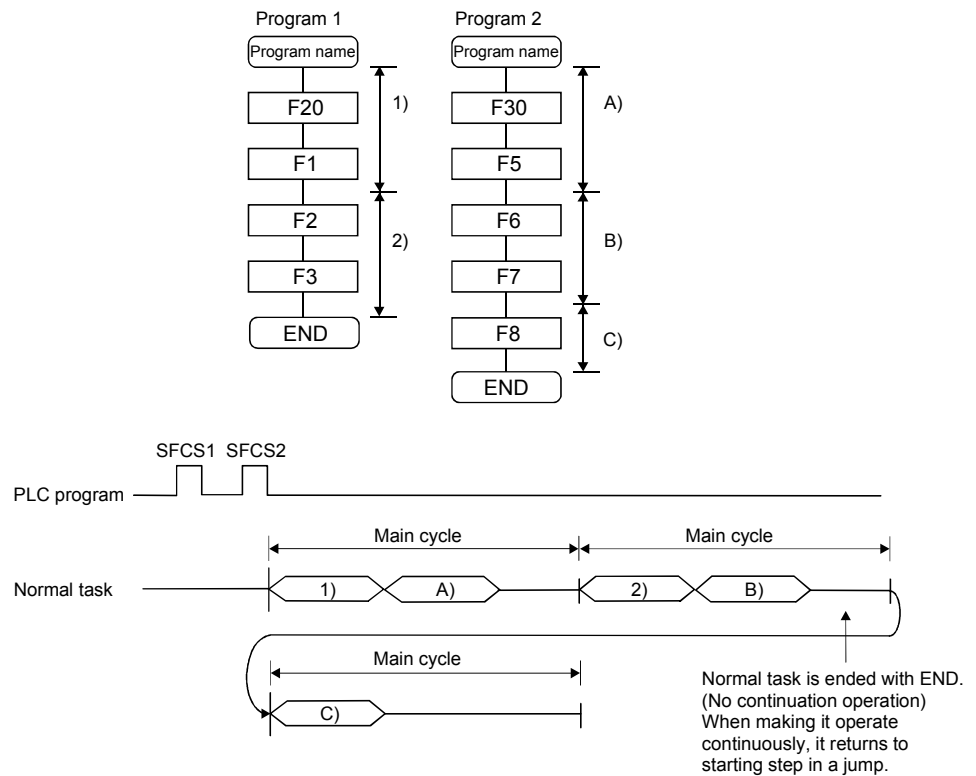
9.2.2 Task operation

(1) Normal task operation

[Operations]

The Motion SFC program is executed in the main cycle (free time) of the Motion CPU.

- Number of consecutive transitions is set to "2".



[Points]

- The Motion SFC program which includes motion control steps should be set to a normal task.
- During execution of an event or NMI task, the execution of the normal task is suspended.
Note that since the normal task allows the event task disable instruction (DI) to be described in an operation control step, the event task can be disabled in the area enclosed by the event task disable instruction (DI) and event task enable instruction (EI).

(2) Event task operation

[Operations]

An event task executes the Motion SFC program at occurrence of an event. There are the following events.

(a) Fixed cycle

The Motion SFC program is executed periodically in any of 0.88ms, 1.77ms, 3.55ms, 7.11ms and 14.2ms cycles.

(b) External interrupt (16 points of I0 to I15)

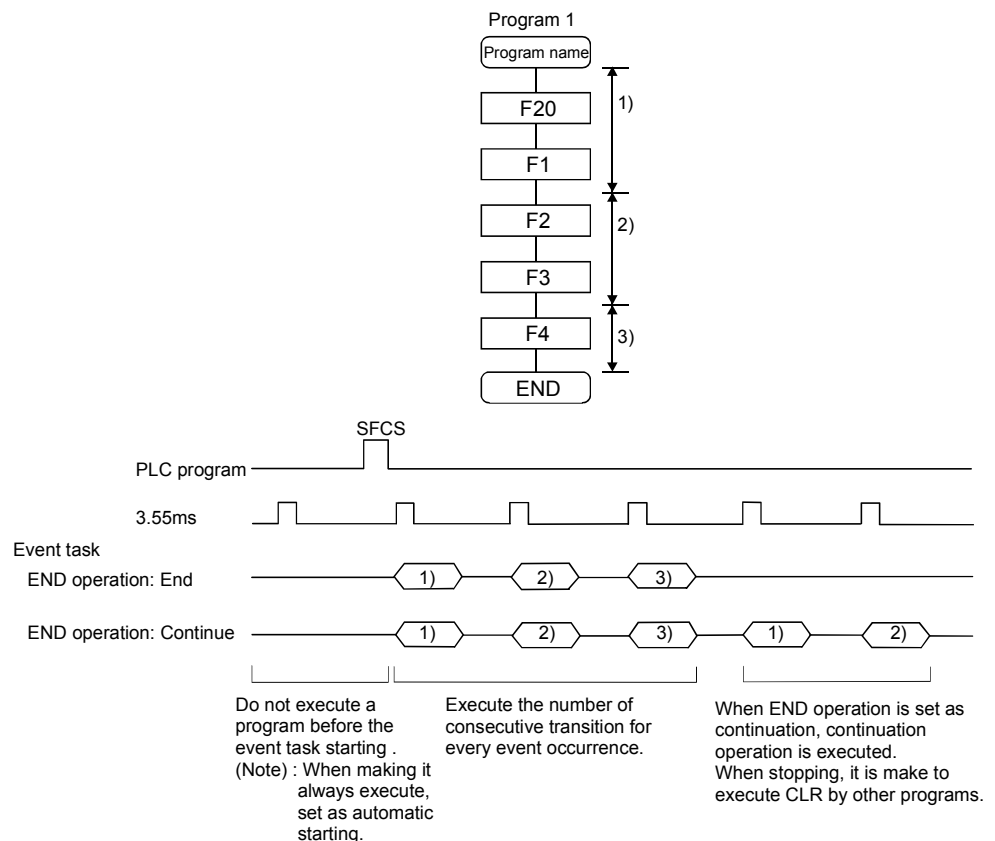
Among 16 points of the QI60 (16-point interrupt module) loaded in the motion slot, the Motion SFC program is run when the input set for an event task turns on.

(c) PLC interrupt

The Motion SFC program is executed when the D(P).GINT instruction is executed in the PLC program.

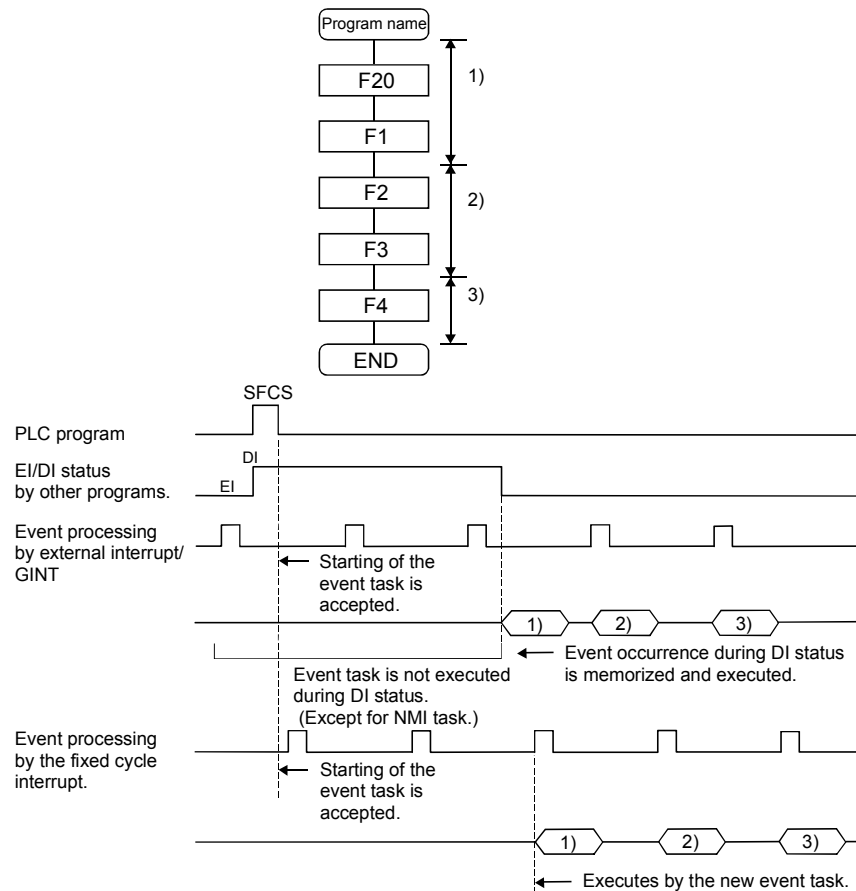
<Example 1> Operation for fixed cycle task (3.55 [ms])

- Number of consecutive transitions is set to "2".



<Example 2> Operation for PLC interrupt by D(P).GINT

- Number of consecutive transitions is set to "2".



[Points]

- Multiple events can be set to one Motion SFC program. However, multiple fixed cycles cannot be set.
- Multiple Motion SFC programs can be executed by one event.
- Motion control steps cannot be executed during the event task.
- The event task cannot be executed when it is disabled by the normal task. The event that occurred during event task disable is executed the moment the event task is enabled.

[Errors]

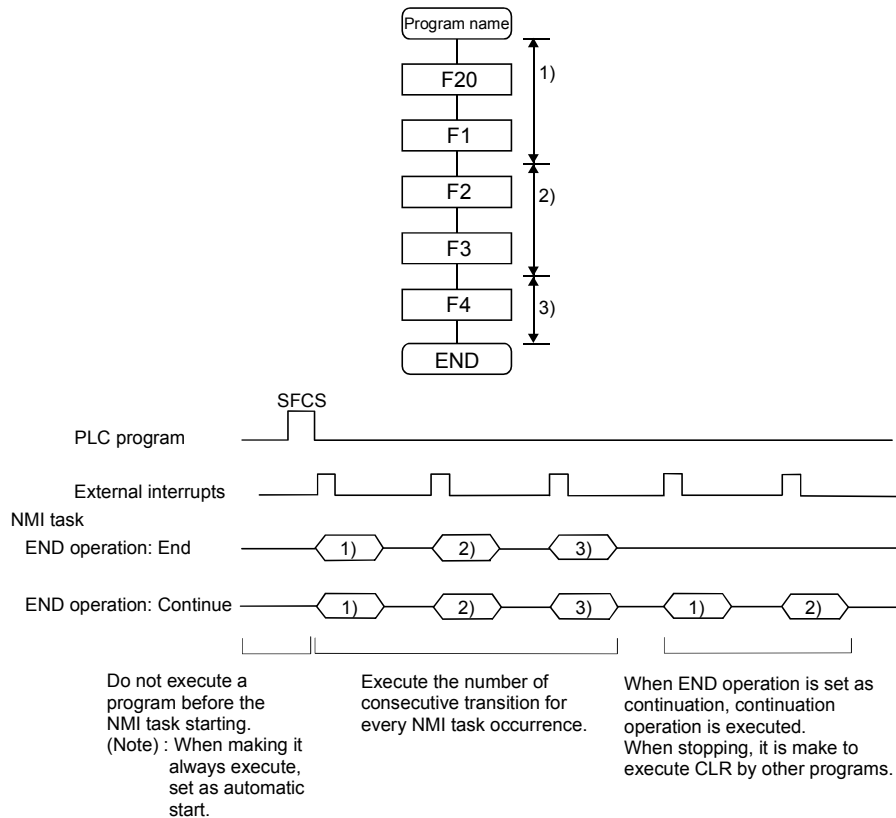
When the motion control step is executed by the Motion SFC program set to the event task, the Motion SFC program error [16113] occurs and stops the Motion SFC program running.

(3) NMI task operation

[Operations]

The Motion SFC program is executed when the input set to the NMI task factor among external interrupts (16 points of QI60) turns on.

- Number of consecutive transitions is set to "2".



[Points]

- NMI task has the highest priority among the normal, event and NMI tasks.
- If the event task is disabled (DI) by the normal task, the interruption of the NMI task is executed, without being masked.

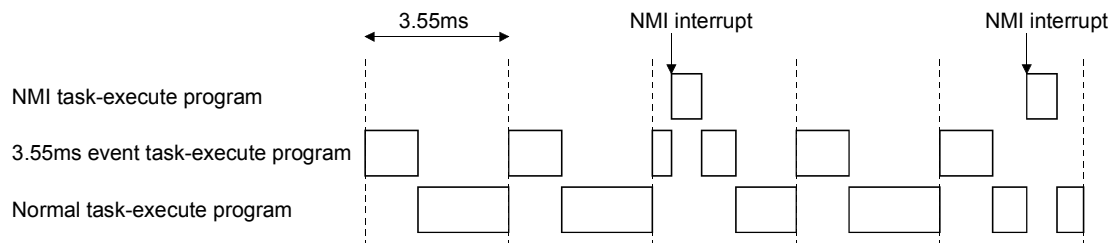
[Errors]

The motion control step is executed during NMI task.

If the motion control step is executed during NMI task, the Motion SFC program error [16113] occurs and stops the Motion SFC program.

9.3 Execution Status of The Multiple Task

Execution status of each Motion SFC program when the Motion SFC program is executed multiple tasks is shown below.



When there are programs which are executed by the NMI task, 3.55ms fixed-cycle even task with a program to run by the NMI task, and the normal task like a chart,

- (1) The 3.55ms fixed-cycle event task is executed at intervals of 3.55ms;
- (2) The NMI task is executed with the highest priority when an NMI interrupt is input; and
- (3) The normal task is executed at free time. as shown above.

[Points]

One Motion SFC program can be executed partially by another task by setting the area to be executed by another task as a subroutine and setting a subroutine running task as another task.

<Example>

No. 0 Main Motion SFC program	Normal task
No. 1 Subroutine	Event task (3.55ms cycle)

⚠ CAUTION

- A normal task may be hardly executed when a NMI task, an event task are executed in many.

9.4 How to Start The Motion SFC Program

The Motion SFC program is executed during PLC ready flag (M2000) is on.
The Motion SFC program may be started by any of the following three methods.

- (1) Automatic start
- (2) Start from the Motion SFC program
- (3) Start from the PLC

Set the starting method in the program parameter for every Motion SFC program.
Refer to Section "9.12 Program Parameters" for parameter setting.

9.4.1 Automatic start

[Operations]

An automatic start is made by turning PLC ready flag (M2000) on.

9.4.2 Start from the Motion SFC program

[Operations]

A start is made by executing a subroutine call/start step in the SFC program.

Refer to "Chapter 4 MOTION SFC PROGRAMS" for details of the subroutine call/start step.

9.4.3 Start from PLC (PLC instruction D(P).SFCS)

The SFC program is started by executing the D(P).SFCS in the PLC program.

Refer to "Chapter 3 MOTION DEDICATED PLC INSTRUCTION" for details.

9.5 How to End The Motion SFC Program

[Operations]

- (1) The Motion SFC program is ended by executing END set in itself.
- (2) The Motion SFC program is stopped by turning off the PLC ready flag (M2000).
- (3) The program can be ended by the clear step.

Refer to Section "4.5.4 Clear step" for details of the clear step.

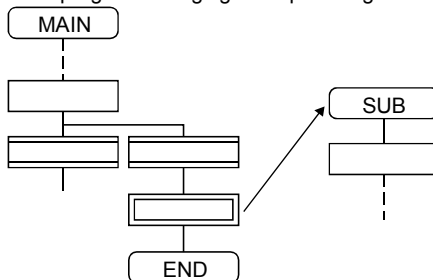
[Point]

- (1) Multiple ENDS can be set in one Motion SFC program.

9.6 How to Change from One Motion SFC Program to Another

Use a subroutine start to stop the Motion SFC program running and switch it to another Motion SFC program.

Motion SFC program changing example using subroutine start



9.7 Operation Performed at Multiple CPU system Power-Off or Reset

When the Multiple CPU system is powered off or reset operation is performed, Motion SFC programs run are shown below.

- (1) When the Multiple CPU system is powered off or reset operation is performed, Motion SFC programs stop to execute.
- (2) At Multiple CPU system power-on or key-reset, the contents of the motion registers #0 to #7999 are held. Initialize them in the Motion SFC programs as required.
- (3) After Multiple CPU system power-on or reset processing, Motion SFC programs run is shown below.
 - The SFC programs set to start automatically are run from the beginning by turning PLC ready flag (M2000) on in the PLC program.
 - The other Motion SFC programs are also executed from the first at starting.

9.8 Operation Performed when CPU is Switched from RUN/STOP

When a RUN/STOP switch is operated, PLC ready flag (M2000) turns on/off in accordance with "Operation at STOP to RUN" of system basic setting.

Refer to Section "3.1.3 Individual parameters" of the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for the details of "Operation at STOP to RUN".

And, refer to the next section for PLC ready flag (M2000) off/on.

9 OPERATION FOR MOTION SFC AND PARAMETER

9.9 Operation Performed when PLC Ready Flag (M2000) Turns OFF/ON

This section explains about the turns off/on of PLC ready flag (M2000).

The on/off condition of PLC ready flag (M2000) differences in "Operation at STOP to RUN" of system basic setting.

Refer to Section "3.1.3 Individual parameters" of the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for details.

[M2000 OFF → ON]

If there is no fault when PLC ready flag (M2000) turns off to on, the PCPU ready flag (SM500) turns on.

When this PCPU ready flag (SM500) turns on, Motion SFC programs can be executed. An automatic start Motion SFC program starts execution from the first.

[M2000 ON → OFF]

When PLC ready flag (M2000) turns off, Motion SFC programs stops to execute and the PCPU ready flag (SM500) turns off.

Since actual outputs PY has whole point turn off.

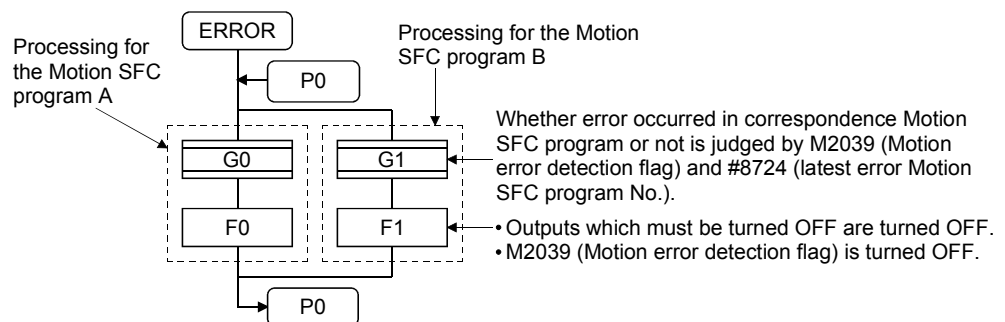
POINT

When the PLC ready flag (M2000) turns off, Motion SFC programs stop but actual outputs PY in the Motion SFC programs do not turn off.

9.10 Operation at The Error Occurrence

Outputs are held if Motion SFC programs stop due to error occurrence.

To turn off outputs at error occurrence, executes the following Motion SFC program.



9 OPERATION FOR MOTION SFC AND PARAMETER

9.11 Task Parameters

No.	Item		Setting item	Initial value	Remark
1	Number of consecutive transitions	Normal task (Normal task common)	1 to 30	3	These parameters are imported at leading edge of PLC ready flag (M2000) and used for control thereafter. When setting/changing the values of these parameters, turns the PLC ready flag (M2000) off.
2	Interrupt setting		Set whether the event task or NMI task is used for external interrupt inputs (I0 to I15).	Event task	

(1) Number of consecutive transitions

[Description]

With "execution of active step → judgment of next transition condition → transition processing performed when condition enables (transition of active step)" defined as a single basic operation of the Motion SFC program execution control in the execution cycle of the corresponding task, this operation is performed for the number of active steps to terminate processing once. And the same operation is processed continuously in the next cycle.

In this case, the transition destination step is executed in the next cycle when the transition condition enables.

Consecutive transition control indicates that transition destination steps are executed one after another in the same one execution cycle when their transition conditions have enabled (single basic operation is performed consecutively).

In this case, the number of consecutive transitions can be set.

Controls in common to the Motion SFC programs executed by normal tasks.

POINT

Set the number of consecutive transitions to the Motion SFC programs executed by event and NMI tasks for every program.

[Errors]

These parameters are imported and checked at leading edge of PLC ready flag (M2000).

When the value that was set is outside the setting range, the following Motion SFC error is set and the initial value is used to control.

Error code (Note)	Error cause		Error processing	Corrective action
	Name	Contents		
17000	Normal task consecutive transition count error	The normal task's consecutive transition count of the Motion SFC program executed by the normal task is outside the range 1 to 30.	The initial value of 3 is used for control.	Turn PLC ready flag (M2000) off, make correction to set the value of within the range, and write it to the CPU.

(Note) : 0 (normal)

(2) Interrupt setting

[Description]

Set whether 16 interrupt input points (I0 to I15) of the QI60 interrupt module loaded in the motion slot are used as NMI or event task inputs.

Setting can be made freely per point.

All points default to event tasks.

[Errors]

None.

9 OPERATION FOR MOTION SFC AND PARAMETER

9.12 Program Parameters

Set the following parameters for every Motion SFC program.

No.	Item	Setting range	Initial value	Remark
1	Start setting	Automatically started or not	Not setting	These parameters are imported at leading edge of PLC ready flag (M2000) and used for control there after. When setting/changing the values of these parameters, turn PLC ready flag (M2000) off.
2	Execute task	It is only one of normal, event and NMI tasks	Normal task	
		When you have set the event task, further set the event which will be enabled. Always set any one of the following 1 to 3. 1. Fixed cycle It is one of 0.88ms, 1.77ms, 3.55ms, 7.11ms and 14.2ms or none. 2. External interrupt (make selection from those set to event task) Multiple interrupt can be set from among I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14 and I15. 3. PLC interrupt Multiple interrupt can be set from among I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14 and I15. The above 1 to 3: Multiple setting is possible. The same event can be shared among multiple Motion SFC programs.	None	
		When you have set the NMI task, further set the interrupt input which will be enabled. 1. External interrupt (make selection from those set to NMI task) Multiple interrupt can be set from among I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14 and I15.		
3	Number of consecutive transitions	1 to 10 Set the number of consecutive transitions toward the program set to the event or NMI task.	1	
4	END operation	End/continue Set the operation mode of the END step toward the program set to the event or NMI task.	End	
5	Executing flag	None/Bit device Set the bit device turned ON while executing Motion SFC program. X0 to X1FFF Y0 to Y1FFF M0 to M8191 B0 to B1FFF U□\G10000.0 to U□\G(10000+p-1).F ^(Note-1) (Self CPU only)	None	

(Note-1): "p" indicates the user setting area points of the Multiple CPU high speed transmission area for the each CPU.

9 OPERATION FOR MOTION SFC AND PARAMETER

POINT
(1) The settings of "END operation" are invalid for the subroutine called program. "END operation" is controlled as "end".
(2) Refer to Chapter 2 of the "Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)" for the user setting area points of the Multiple CPU high speed transmission area.

(1) Start setting

[Description]

The following control is changed by "automatically started or not" setting.

- Program run by normal task

No.	Item	When "automatically started"	When "not automatically started"
1	Start control	In the main cycle after PLC ready flag (M2000) ON, the program is executed from the initial (first) step in accordance with the number of consecutive transitions of the normal task.	<p>The program is started by the Motion SFC start instruction (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from the Motion SFC program.</p> <ul style="list-style-type: none"> • When started by the (D(P).SFCS) instruction In the main cycle after execution of the (D(P).SFCS) instruction, the program is executed from the initial (first) step in accordance with the number of consecutive transitions of the normal task. • When subroutine started In the (next) main cycle after execution of GSUB, the program is executed from the first step in accordance with the number of consecutive transitions of the normal task. • When subroutine called The program is executed in the same cycle from the first step.
		After that, the program is executed continuously by the number of consecutive transitions of the normal task in the Motion CPU main cycle. (The settings of "executed task" and "number of consecutive transitions" of the subroutine called program are invalid. It is controlled as the normal task.)	
2	END control END	<p>Ends the self program.</p> <p>Again, the program is started by the Motion SFC start instruction (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from the Motion SFC program.</p>	

9 OPERATION FOR MOTION SFC AND PARAMETER

• Program run by event task

No.	Item	When "automatically started"	When "not automatically started"
1	Start control	At occurrence of a valid event after PLC ready flag (M2000) ON, the program is executed from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program.	<p>The program is started by the Motion SFC start instruction (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from within the Motion SFC program.</p> <ul style="list-style-type: none"> • When started by the D(P).SFCS instruction At occurrence of a valid event after execution of the D(P).SFCS instruction, the program is run from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program. • When subroutine started At occurrence of a valid event after execution of GSUB, the program is executed from the first step in accordance with the number of consecutive transitions of the corresponding program. • When subroutine called The program is executed immediately from the first step.
		After that, the program is executed continuously by the number of consecutive transitions of the corresponding program at occurrence of a valid event. (The subroutine called program is controlled in accordance with the "executed task" and "number of consecutive transitions" of the call source program.)	
2	END control END	As specified for END operation.	

• Program run by NMI task

No.	Item	When "automatically started"	When "not automatically started"
1	Start control	At occurrence of a valid event after PLC ready flag (M2000) ON, the program is executed from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program.	<p>The program is started by the Motion SFC start instruction (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from within the Motion SFC program.</p> <ul style="list-style-type: none"> • When started by the D(P).SFCS instruction At occurrence of a valid event after execution of the D(P).SFCS instruction, the program is run from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program. • When subroutine started At occurrence of a valid event after execution of GSUB, the program is executed from the first step in accordance with the number of consecutive transitions of the corresponding program. • When subroutine called The program is executed immediately from the first step.
		After that, the program is executed continuously by the number of consecutive transitions of the corresponding program at occurrence of a valid event.	
2	END control END	As specified for END operation.	

[Errors]

None.

POINT

In the case of the program which is executed by the normal task, write the program so that it is not ended by **END** but it returns to the starting step by a jump when starting of the automatically from an initial again.

(2) Execute task

[Description]

Set the timing (task) to execute a program.

Specify whether the program will be run by only one of the "normal task (main cycle), event task (fixed cycle, external interrupt, PLC interrupt) and NMI task (external interrupt)".

When the event task is set, multiple events among the "fixed cycle, external interrupt (for event task) and PLC interrupt".

However, multiple fixed cycles cannot be set toward one Motion SFC program.

<Example> Interrupt setting: Inputs for event task I6, I7, I8, I9, I10, I11, I12, I13, I14 and I15

Motion SFC program No. 10 – event : Fixed cycle (3.55ms)

Motion SFC program No. 20 – event :

Fixed cycle (1.77ms) + external interrupt (I6)

Motion SFC program No. 30 – event :

External interrupts (I7, I15) + PLC CPU interrupt

When the NMI task is set, multiple interrupt inputs among the external interrupts (for NMI task) can be set.

<Example> Interrupt setting: Inputs for NMI task I0, I1, I2, I3, I4, I5

Motion SFC program No. 10 – NMI : I0

Motion SFC program No. 20 – NMI : I1 + I2

Motion SFC program No. 30 – NMI : I5

[Errors]

This program parameter is imported at leading edge of PLC ready flag (M2000), and is checked at starting of the Motion SFC program (automatic start, start from PLC or subroutine start).

When the value is illegal, either of the following Motion SFC errors is set and the initial value is controlled.

Error code (Note)	Error cause		Error processing	Corrective action
	Name	Contents		
17010	Execute task setting is illegal	Multiple events among the normal, event and NMI tasks are set, or one is not set.	The initial value (normal task) is controlled.	Turn PLC ready flag (M2000) off, make correction to set the value of within the range, and write it to the CPU.
17011	Executed task setting is illegal (event)	Two or more fixed cycles of the event task have been set.		

(Note) : 0 (normal)

POINT

Since the execute task can be set for every Motion SFC program No., multiple programs need not be written for single control (machine operation) to divide execution timing-based processing's.

For example, it can be achieved easily by subroutine starting the areas to be run in fixed cycle and to be run by external interrupt partially in the Motion SFC program run by the normal task.

(3) Number of consecutive transitions

[Description]

Set the number of consecutive transitions to program executed by the event or NMI task for every program.

Refer to Section "9.11 Task Parameters" for number of consecutive transitions.

[Errors]

This program parameter is imported at leading edge of PLC ready flag (M2000), and is checked at starting of the Motion SFC program (automatic start, start from PLC or subroutine start).

When the value is illegal, either of the following Motion SFC errors is set and the initial value is controlled.

Error code (Note)	Error cause		Error processing	Corrective action
	Name	Contents		
17001	Event task consecutive transition count error	The number of consecutive transitions of the Motion SFC program started by the event task is outside the range 1 to 10.	The initial value of 1 is controlled.	Turn PLC ready flag (M2000) off, make correction to set the value of within the range, and write it to the CPU.
17002	NMI task consecutive transition count error	The number of consecutive transitions of the Motion SFC program started by the NMI task is outside the range 1 to 10.		

(Note) : 0 (normal)

(4) END operation

[Description]

Set the operation at execution of the END step toward the program executed by the event or NMI task.

This varies the specifications for the following items.

- Program run by NMI task

No.	Item	When "ended"	When "continued"
1	Control at END execution	Ends the self program.	Ends to execute the self program with this event/interrupt.
2	Restart after END execution	Again, the program is started by the Motion SFC start instruction (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from the Motion SFC program.	Restarted at occurrence of the next event/interrupt, and run from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program. After that, at occurrence of an event/interrupt, the program is executed in accordance with the number of consecutive transitions of the corresponding program.
3	Restart after end by clear step CLR	Again, the program is started by the Motion SFC start instruction (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from the Motion SFC program.	

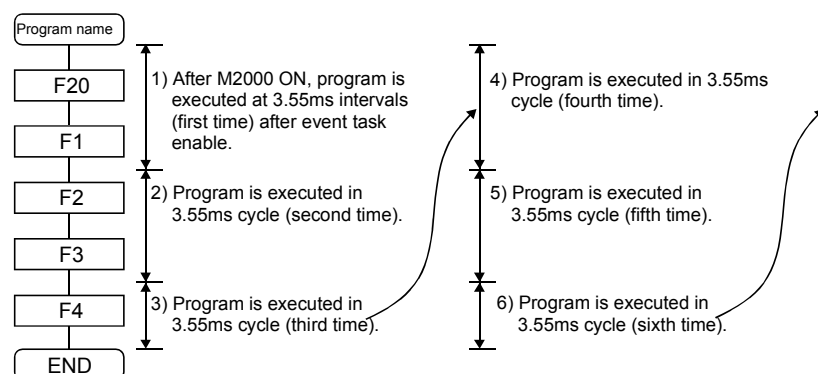
POINT

The END operation of subroutine called program is controlled as an "end".

- The following operation example assumes that the END operation is "continued."

Program parameters

- Automatically started
- Execute task = event 3.55ms
- Number of consecutive transitions = 2
- End operation "continued"



(5) Executing flag

The set bit device turns ON by Motion SFC program start, and it turns OFF by program end.

[illegible]

10. ONLINE CHANGE IN THE MOTION SFC PROGRAM

10.1 Online Change in The Motion SFC Program

This function is used to write to the Motion SFC program to the SRAM built-in Motion CPU during the positioning control (7-segment LED : Steady "RUN" display). Program correction and a check of operation can be executed repeatedly at the Multiple CPU system start.

Data in which online change is possible are shown below.

Applicable data		Online change	Remarks
System setting data	System setting	×	
	Servo setting data	×	
Motion SFC program	Motion SFC parameter	×	
	Motion SFC chart	○	Online change is possible for the only program during stop.
	Operation control step (F/FS)	○	
	Transition (G)	○	
	Servo program (K)	○	Online change of mode assignment setting is not possible.
Mechanical system program (SV22)		×	
Cam data (SV22)		×	

○ : Possible × : Not possible

POINT	
(1)	Program writing is executed during the positioning control in the online change. Be safely careful enough for work.
(2)	Programs writing to the SRAM built-in Motion CPU at the mode operated by ROM in the online change. If the online change is executed at the mode operated by ROM, it returns to the contents of program written in the FLASH ROM built-in Motion CPU by the next power ON or reset of the Multiple CPU system.
(3)	If the online change is executed simultaneously to one Motion CPU from the multiple personal computers, a program writing may not be executed. Please do not perform.
(4)	<p>If the online changes are executed by other personal computer during the following operation by MT Developer, injustice of a monitor value and operation failure may occur. Please do not perform.</p> <ul style="list-style-type: none"> • Monitor mode of the Motion SFC program • Debug mode of the Motion SFC program • Test mode
(5)	If the online change of Motion SFC chart added newly is executed, since the online change of Motion SFC parameter cannot be executed, it operates as the normal task (default value).
(6)	When using the SV22, if the online change is executed by changing the "program/servo program editor screen – [Mode assignment setting]", the contents of change are not reflected.
(7)	If the cables between the personal computer and PLC CPU module fall out, or the power supply of the Multiple CPU system turns OFF or resets, the program is corrupted.
	Write the program again with "Write to CPU" screen of MT Developer.

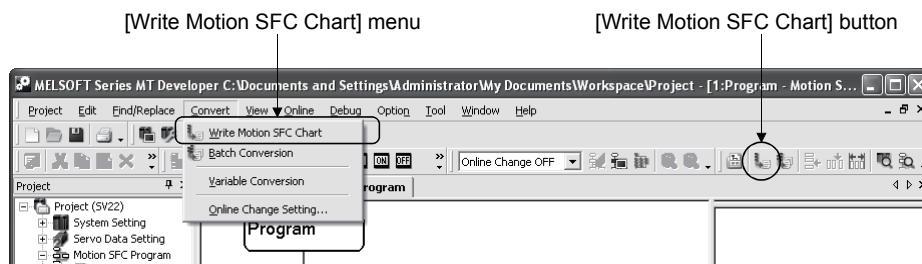
10.1.1 Operating method for the online change

Select the "Online change OFF/ON" of Motion SFC program with the "Motion SFC program editor screen [Convert] menu – [Online Change Setting]" of MT Developer.

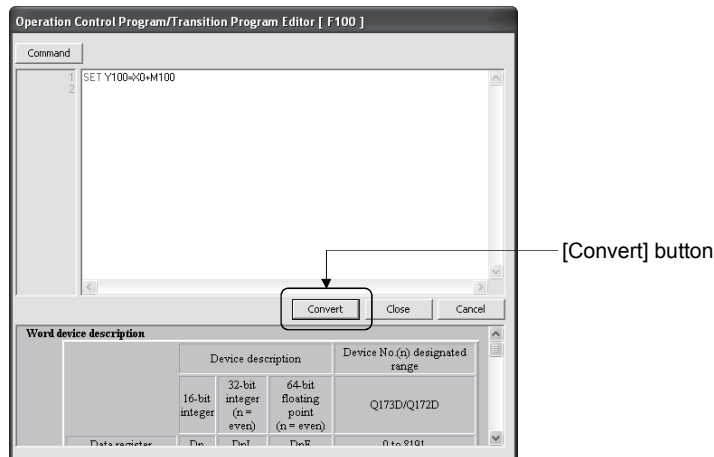
There are following three methods for the online change of Motion SFC program.

- When the Motion SFC program editor screen [Write Motion SFC Chart] is used ---- Online change of the Motion SFC chart
- When the operation control/transition program editor screen [Convert] is used ---- Online change of the operation control/transition program editor screen
- When the servo program editor screen [Convert] is used ---- Online change of the servo program

- (1) When the Motion SFC program editor screen [Write Motion SFC Chart] is used.
Online change of the Motion SFC program during edit is executed by selecting the [Write Motion SFC Chart] button or menu.
Online change is possible to the Motion SFC program during stop.
If the online change is made to the program during execution, an alarm message indicates. (Execution/stop state of the Motion SFC program can be checked with the program batch monitor.)
If the start request is made to the program during online change, the Motion SFC start error (error code 16007: online change) will occur and the program does not start.



- (2) When the operation control/transition program editor screen [Convert] is used.
 Online change of the operation control/transition program during edit is executed by selecting the [Convert] button.
 Online change is possible to the operation control/transition program during execution.
 A program that the online change was made is executed from the next scan.

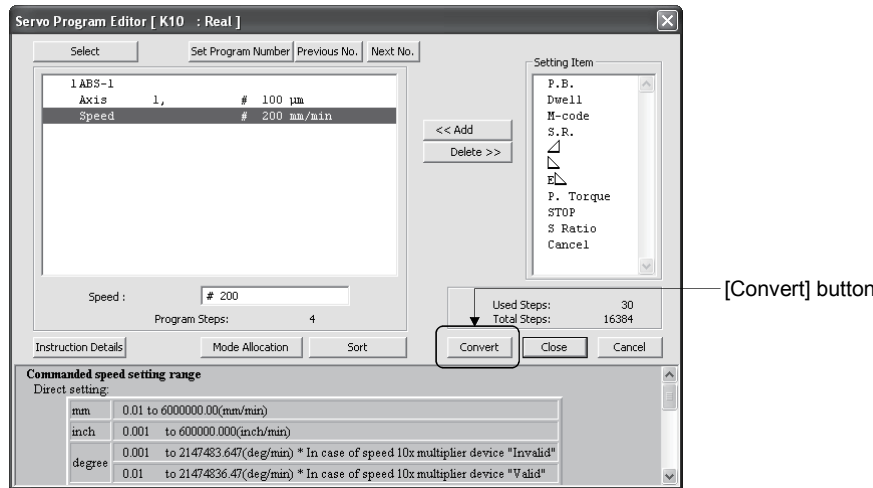


Operations for which made the online change to the operation control/transition program during execution in the following conditions are shown below. Be careful to execute the online change in the following conditions.

Program	Condition	Operation
<pre> graph TD FSn[FSn] --> Gn[Gn] Gn --> FSn </pre> or <pre> graph TD FSn[FSn] --> Gn[Gn] Gn --> FSn </pre>	<ul style="list-style-type: none"> Online change of the FSn operation control program is executed during FSn execution in the state of waiting for the completion of condition for Gn. 	<ul style="list-style-type: none"> After completion of online change, the FSn repeats the operation control program that the online change was made until the completion of condition for Gn.
<pre> graph TD Gn[Gn] --> Gn Gn --> Gn </pre> or <pre> graph TD Gn[Gn] --> Gn Gn --> Gn </pre>	<ul style="list-style-type: none"> Online change of the Gn program is executed in the state of waiting for the completion of condition for Gn. (The conditional sentences of program to write are except the TIME instruction.) 	<ul style="list-style-type: none"> After completion of online change, the Gn does not transit to the next step until the completion of condition for program that the online change was made.
<pre> graph TD Gn[Gn] --> Gn Gn --> Gn </pre> or <pre> graph TD Gn[Gn] --> Gn Gn --> Gn </pre>	<ul style="list-style-type: none"> Online change of the Gn program including the TIME instruction is executed in the state of waiting for the completion of condition for Gn. 	<ul style="list-style-type: none"> After completion of online change, Gn is ended regardless of the waiting time of TIME instruction and the next step is executed.
<pre> graph TD Kn[Kn] --> Gn[Gn] Gn --> Kn </pre> or <pre> graph TD Kn[Kn] --> Gn[Gn] Gn --> Kn </pre>	<ul style="list-style-type: none"> Online change of the Gn program during the servo program execution for Kn. 	<ul style="list-style-type: none"> After execution of servo program, the program of changed Gn is executed.

- (3) When the servo program editor screen [Convert] is used.
Online change of the servo program during edit is executed by selecting the [Convert] button.

Online change is possible to the servo program during execution.
A program that the online change was made is executed at the next servo program start.



Operations for which made the online change to the servo program in the following conditions during execution are shown below. Be careful to execute the online change in the following conditions.

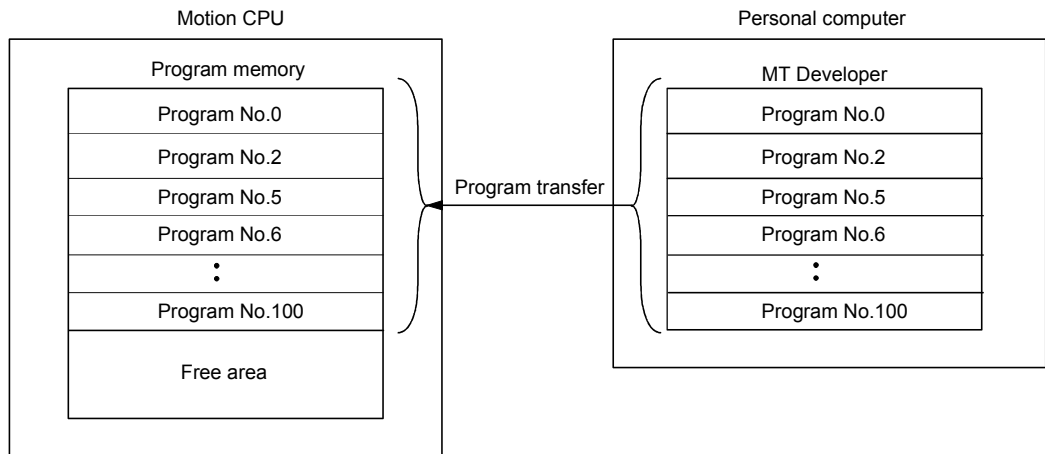
Program	Condition	Operation
<div>ON bit device</div> <div>Kn</div> <div>or</div> <div>OFF bit device</div> <div>Kn</div>	<ul style="list-style-type: none">Online change of the servo program Kn at the WAITON or after WAITOFF is executed in the state of waiting for the completion of condition for WAITON/WAITOFF.	<ul style="list-style-type: none">After completion of condition for WAITON/WAITOFF, the servo program before the online change is started.The servo program that the online change was made is executed at the next servo program start.
<div>Gn</div> <div>Kn</div> <div>or</div> <div>Gn</div> <div>Kn</div>	<ul style="list-style-type: none">Online change of the servo program Kn after Gn is executed in the state of waiting for the completion of condition for Gn.	<ul style="list-style-type: none">After completion of condition for Gn, the servo program that online change was made is executed.

10.1.2 Transfer of program

The outline operations to transfer the program from MT Developer to the program memory of Motion CPU are described.

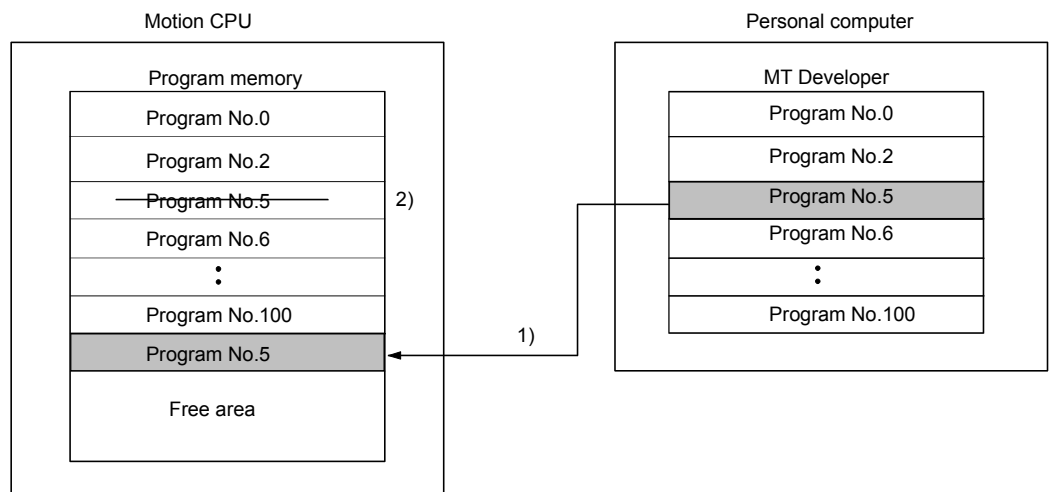
(1) Program writing by [Online] menu - [Write to CPU]

- (a) After transfer, programs are stored in the program memory of Motion CPU stuffing to the front for every kind.

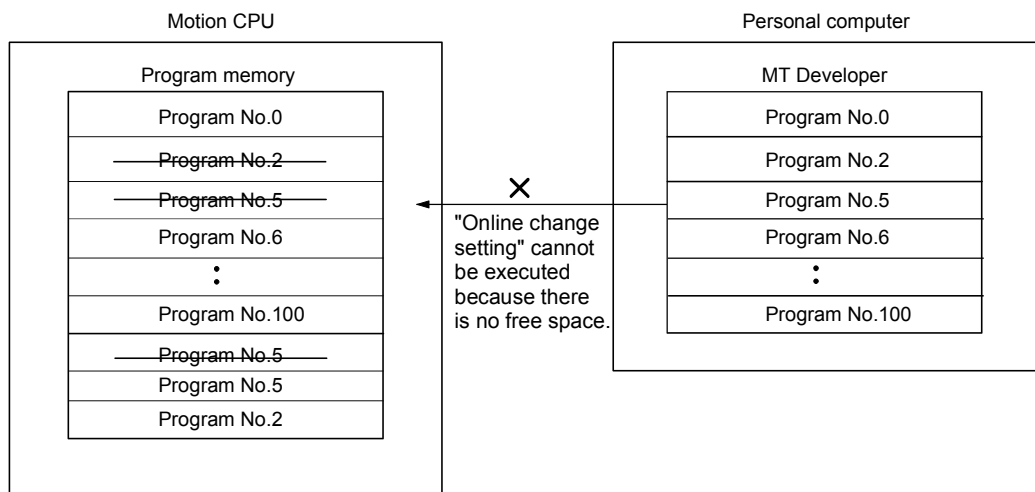


(2) Program writing by the [Online change]

- (a) After online change, a program to execute the online change is stored in the free area after the program stored previously. (Refer to 1))
After that, the program written in previously is made invalid and the new program is made valid. (Refer to 2))



- (b) If the online change is executed repeatedly, the free space in program memory is lost and the online change may not be executed. In this case, an error message is displayed by MT Developer at the online change, and "Online change OFF" is set.



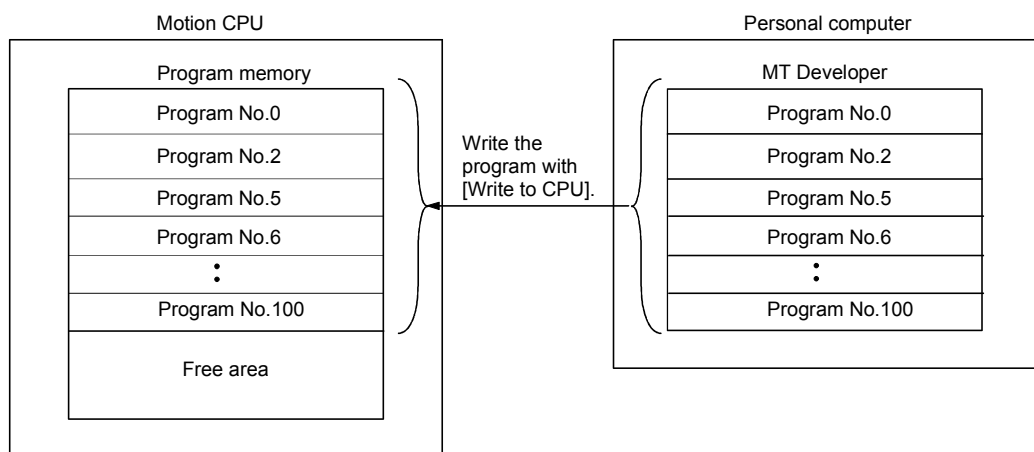
- (c) In the case of b), arrange to stuff to the front the invalid programs.

Operation procedures to stuff to the front are shown below.

- 1) Select the followings by Motion SFC program editor screen of MT Developer.

- For Motion SFC chart and operation control/transition program
Select [Convert] menu - [Batch Conversion].
- For Servo program
Select [Tool] menu - [Sort of Servo Program].

- 2) Execute the program writing with [Online] menu – [Write to CPU] in the stop state of Motion CPU.



MEMO

[illegible]

11. USER FILES

User file list and directory structure is shown below

11.1 Project

Every user file is managed in a workspace, and multiple projects can be stored in a single workspace. "Workspace name" folder and "Project name" folder are created when initially saving a project or when copying a project as indicated on the next page.

POINT

- "Workspace path name" + "Workspace name" + "Project name" are restricted to 200 characters in length.

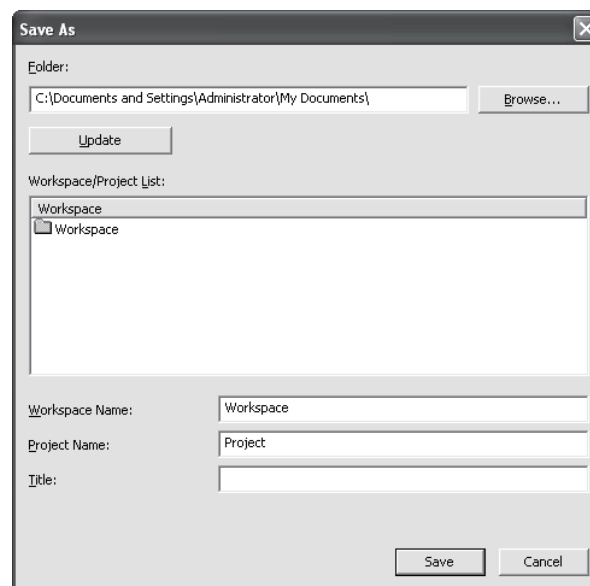
<Example>

C:\Documents and Settings\Administrator\My Documents\Workspace\Project

Workspace path name

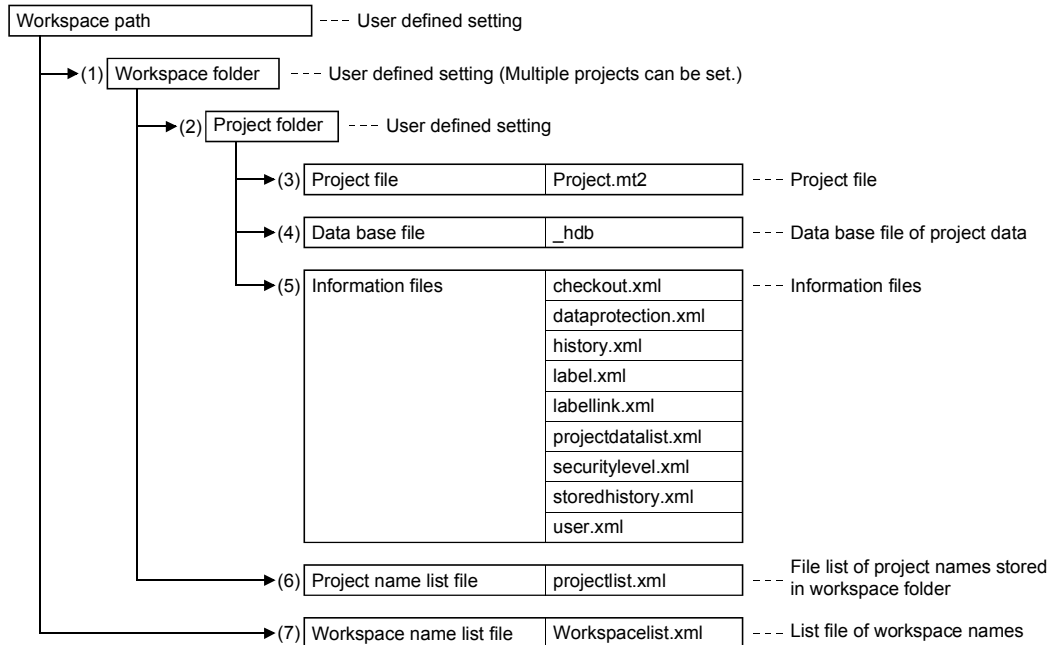
Workspace name

Project name



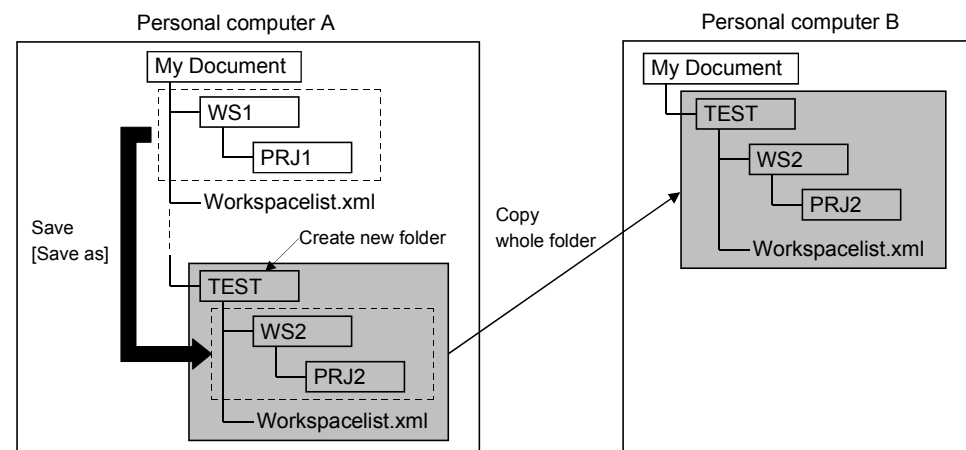
11.2 User File List

User file list is shown below.



POINT

- (1) Double-clicking on the "Project file (Project.mt2)" opens a project.
- (2) "Workspace" folder and "Workspace list file (Workspacelist.xml)" that composes a system should be stored in the same workspace path folder.
- (3) The procedure for using a project stored in "Workspace" folder on another PC is shown below.
 - 1) Create any folder (ex. TEST) on PC "A".
 - 2) Open a project from MT Developer.
 - 3) Save ([Save as] menu) the project in a folder created by procedure 1).
 - 4) Close MT Developer.
 - 5) In Windows explorer, copy the whole saved folder to electronic media (USB memory etc.).
 - 6) Open the project on PC "B" from MT Developer.



12. ERROR CODE LISTS

When an error occurs while the Motion CPU is running, the error information is stored in the error history devices (#8640 to #8735), special relay (SM) and special register (SD).

12.1 Reading Procedure for Error Codes

When an error occurs while the Motion SFC program is operating, the error code and error message can be read using MT Developer.

The procedure for reading error codes using MT Developer is shown below.

- (1) Connect the PLC CPU module to personal computer (IBM PC/AT).
- (2) Start MT Developer.
- (3) Select [Online] - [Read from CPU] Menu of MT Developer, and read the project data from Motion CPU.
- (4) Start the monitor screen of MT Developer and select [Motion CPU error batch monitor] menu.
- (5) Confirm the error codes and error messages displayed on screen.

Refer to help of MT Developer for details of operating method.

As for the self-diagnosis error code, confirmation can be done by the PC diagnosis of GX Developer.

Refer to the GX Developer operation manual for the GX Developer operation procedure.

12.2 Motion Error Related Devices

(1) Motion error history devices (#8640 to #8735)

Eighth in the past (Seventh in the past to latest) error information are stored as a history. #8724 to #8735 are latest errors.

All errors, including the Motion SFC control errors and the conventional minor, major, servo, servo program setting and mode changing errors are stored. At error occurrence, the "Motion error detection flag (M2039)" is also set.

Motion error history is backed-up.

Use the Motion error history clear request flag (M2035) or

MT Developer to clear the Motion error history.

Also, the self-diagnostic errors of error code 10000 or less are stored in the Motion error history.

Refer to the following manuals for details of each error.

Error type	Reference manual
• Minor error	Q173DCPU/Q172DCPU Motion controller (SV13/SV22)
• Major error	Programming Manual (REAL MODE)
• Servo error	Q173DCPU/Q172DCPU Motion controller (SV22) Programming
• Servo program setting error	Manual (VIRTUAL MODE)
• Self-diagnosis error	Q173DCPU/Q172DCPU Motion controller Programming Manual (COMMON)

Table 12.1 Motion error history device

Error information								Signal name
Seventh in past	Sixth in past	Fifth in past	Fourth in past	Third in past	Second in past	First in past	Latest	
#8640	#8652	#8664	#8676	#8688	#8700	#8712	#8724	Error Motion SFC program No.
#8641	#8653	#8665	#8677	#8689	#8701	#8713	#8725	Error type
#8642	#8654	#8666	#8678	#8690	#8702	#8714	#8726	Error program No.
#8643	#8655	#8667	#8679	#8691	#8703	#8715	#8727	Error block No./Motion SFC list/ Line No./Axis No.
#8644	#8656	#8668	#8680	#8692	#8704	#8716	#8728	Error code
#8645	#8657	#8669	#8681	#8693	#8705	#8717	#8729	Error occurrence time (Year/month)
#8646	#8658	#8670	#8682	#8694	#8706	#8718	#8730	Error occurrence time (Day/hour)
#8647	#8659	#8671	#8683	#8695	#8707	#8719	#8731	Error occurrence time (Minute/second)
#8648	#8660	#8672	#8684	#8696	#8708	#8720	#8732	Error setting data information
#8649	#8661	#8673	#8685	#8697	#8709	#8721	#8733	Unusable
#8650	#8662	#8674	#8686	#8698	#8710	#8722	#8734	Error setting data
#8651	#8663	#8675	#8687	#8699	#8711	#8723	#8735	

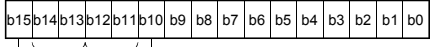
The contents of Motion error history device error information are shown in Table 12.2.

12 ERROR CODE LISTS

Table 12.2 Motion error history device error information

Signal name		Description	
		Motion SFC control errors	Conventional errors
Error Motion SFC program No.		0 to 255 : Motion SFC program No. in error -1 : Independent of Motion SFC program	-1
Error type		20 : F/FS 21 : G 22 : K or other (not any of F/FS, G and SFC chart) 23 : Motion SFC chart	3 : Minor/major error 4 : Minor/major error (virtual servomotor shaft) (SV22) 5 : Minor/major error (synchronous encoder shaft) (SV22) 6 : Error detected in the servo amplifier 7 : Servo program setting error 8 : Mode change error (SV22) 9 : Manual pulse generator axis setting error 10 : Test mode request error 11 : WDT error 13 : Self-diagnostic error (Error code: 10000 or less) 14 : System setting error
Error program No.		0 to 4095 : F/FS, G, K program No. 0 to 255 : GSUB program No. -1 : Independent of F/FS, G, K, GSUB	<ul style="list-style-type: none"> Error type: "3", "4" or "7" 0 to 4095 : Servo program No. FFFFH : JOG operation FFFEH : Manual pulse generator FFFDH : Test mode (Home position return, servo diagnosis, servo startup) FF00H : Others Error type: except "3", "4" or "7" -1
Error block No./ Motion SFC list line No./axis No.		0 to 8191 : F/FS or G program's block No. (line No.) when error type is "1" or "2" 0 to 8188 : Motion SFC list line No. when error type is "-2" -1 : Independent of block when error type is "-1" or error type is "1" or "2"	1 to 32 : Corresponding axis No. when error type is any of "3" to "6" -1 : Others
Error code		16000 and later (Refer to Chapter "12 ERROR CODE LISTS".)	<ul style="list-style-type: none"> Conventional error code (less than 16000) when error type is any of "3" to "6" Error code stored in SD517 when error type is "7" Error code stored in SD504 when error type is "8" -1 when error type is "9" or "10" Error code stored in SD512 when error type is "11" Error code stored in SD0 when error type is "13" or "14".
Error occurrence time	Year/ month Day/ hour Minute/ second	The clock data at error occurrence (SD210, SD211, SD212) are set. (BCD code, year in its lower 2 digits)	

12 ERROR CODE LISTS

Signal name	Description	
	Motion SFC control errors	Conventional errors
Error setting data information	0 : None (Fixed at 0)	 <p>Speed control 10× multiplier setting for degree axis 0 : Invalid 1 : Valid The status at error occurrence is set when the unit of error setting data is set to "11: Control unit (speed data)" and the control unit is set to "10: degree".</p> <p>Control unit 00 : mm 01 : inch 10 : degree 11 : PLS • Virtual servo motor axis : "11: Fixed at PLS" • Valid when only the unit of error setting data is valid</p> <p>Unit of error setting data 00 : None 01 : Axis unit, Output module unit (At output module error occurrence) 10 : Control unit (Address data, Radius-specified allowable error range for circular interpolation) 11 : Control unit (Speed data)</p> <p>Error setting data 0 : No data 1 : Data</p>
Unusable	—	—
Error setting data	0 : None (Fixed at 0)	Setting data in error cause

(2) Motion error detection flag (M2039)

(Refresh cycle : Scan time)

The Motion error detection flag (M2039) turns on when any of the errors detected by the Motion CPU occurs.

At error occurrence, data are set to the error devices in the following procedure.

- Set the error code to each axis or error devices.
- Turns on the error detection signal of each axis or error.
- Set the error information to the above "Motion error history devices (#8640 to #8735)".
- Turns on the Motion error detection flag (M2039) .

In the user program, reset the "Motion error detection flag (M2039)" after reading the error history at the "Motion error detection flag (M2039)".

After that, "Motion error detection flag (M2039)" turns on again at occurrence of a new error.

POINT

- Resetting the "Motion error detection flag (M2039)" will not reset (clear to zero) the "Motion error history devices (#8640 to #8735)".
After power-on, they always controls the error history continuously.
- Set the clock data and clock data read request (SM801) in the user program.

12 ERROR CODE LISTS

12.3 Motion SFC Error Code List

(1) Motion SFC program start errors (16000 to 16099)

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
16000	PLC ready OFF (SFCS)	• At a start by D(P).SFCS instruction, PLC ready flag (M2000) or PCPU ready flag (SM500) is OFF.	The specified Motion SFC program does not start.	Provide ON of the PLC ready flag (M2000) and PCPU ready flag (SM500) as start interlocks.
16001	Motion SFC program No. error (SFCS)	• At a start by D(P).SFCS instruction, the range of 0 to 255 is specified in the Motion SFC program No..		Check the Motion SFC program No., and correct a PLC program.
16002	None Motion SFC program (SFCS)	• At a Motion SFC program start by D(P).SFCS instruction, the specified Motion SFC program does not exist.		Check the Motion SFC program No., and correct a PLC program, or create the non-created Motion SFC program.
16003	Double start error	• At a Motion SFC program start by D(P).SFCS instruction, the same Motion SFC program starts.		Double start should be managed on the user side. Provide the user's starting signal as a start interlock in the PLC program.
16004	PLC ready OFF (GINT)	• D(P).GINT instruction was executed with PLC ready flag (M2000) or PCPU ready flag (SM500) is OFF.	The active step of Motion SFC program executed by "PLC interrupt" is not processed. "Interrupt instruction No." is set in the error Motion SFC program No..	Provide ON of PLC ready flag (M2000) and PCPU ready flag (SM500) as D(P).GINT execution interlocks.
16005	None Motion SFC program	• At a Motion SFC program start by automatic start setting or GSUB, the specified Motion SFC program does not exist.	The specified Motion SFC program does not start. When it started by GSUB, the start source Motion SFC also stop to execute.	Check the Motion SFC program No., and correct a program, or create the non-created Motion SFC program.
16006	Double start error	• At a Motion SFC program start by automatic start setting or GSUB, the same Motion SFC program is already starting.		Double start should be managed on the user side. Provide the user's starting signal as an interlocks in the transition condition.
16007	Online change	• The Motion SFC program which is rewriting the Motion SFC chart by online change was started.	The specified Motion SFC program does not start.	Start after the completion of online change.

12 ERROR CODE LISTS

(2) Motion SFC interpreter detection errors (16100 to 16199)

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
16100	Motion SFC program error (grammatical error)	<ul style="list-style-type: none"> The code exists but is grammatically erroneous. Though not within branch-coupling, a label/jump code within selective branch-coupling or a label/jump code within parallel branch-coupling exists. 	<p>Stop to execute the applicable Motion SFC program No..</p> <p>For the subroutine called program, the call source program also stops to execute.</p>	<p>The Motion SFC program code is corrupted.</p> <p>Turn PLC ready flag (M2000) OFF and write the Motion SFC program again.</p> <p>Or, replace the external battery if it passed over a life.</p>
16101		<ul style="list-style-type: none"> Selective branch destinations are all headed by other than SFT or WAIT transitions. 		
16102		<ul style="list-style-type: none"> WAITON/WAITOFF is not followed by a motion control step. (However, this is permitted to a pointer (Pn) or jump (Pn).) 		
16103		<ul style="list-style-type: none"> A parallel branch is followed by an END step without a parallel coupling. 		
16104	Motion SFC code error	<ul style="list-style-type: none"> An impossible code is used. The internal code is corrupted. 		
16105	Jump code error 1	<ul style="list-style-type: none"> Internal code (list code) error in jump destination information 		
16106	Jump code error 2	<ul style="list-style-type: none"> Internal code (label information) error in jump destination information 		
16107	Jump code error 3	<ul style="list-style-type: none"> Internal code (label No.) error in jump destination information 		
16108	Jump code error 4	<ul style="list-style-type: none"> Internal code (label address) error in jump destination information 		
16109	Jump destination error	<ul style="list-style-type: none"> The specified pointer does not exist at the jump destination. 		
16110	GSUB setting error 1	<ul style="list-style-type: none"> The self program was called/started by GSUB. 		<p>GSUB cannot call its own or main program.</p> <p>Correct the Motion SFC program.</p>
16111	GSUB setting error 2	<ul style="list-style-type: none"> The main program was called/started by GSUB. 		
16112	Parallel branch nesting excess	<ul style="list-style-type: none"> Nesting of parallel branches within a parallel branch route exceeded four levels. 		<p>The nesting of parallel branch is up to four levels.</p> <p>Subroutine the branch destination processing and correct the program.</p>
16113	Executed task error	<ul style="list-style-type: none"> An attempt was made to execute a motion control step K with an event or NMI task. 		<p>Motion control steps cannot be executed in the Motion SFC programs executed by the event and NMI tasks.</p>
16120	Simultaneously active step count excess	<ul style="list-style-type: none"> The number of simultaneously active steps exceeded 256 during execution. 		<p>Number of simultaneously active steps is maximum 256. Re-examine the Motion SFC program.</p>

12 ERROR CODE LISTS

(3) Motion SFC program run errors (16200 to 16299)

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
16200	No specified program (Kn)	• The servo program (Kn) specified with the motion control step does not exist.	<p>Stop to execute the applicable Motion SFC program No..</p> <p>For the subroutine called program, the call source program also stops to execute.</p>	Create the specified servo program.
16201	No specified program (Fn/FSn)	• The operation control program (Fn/FSn) specified with the operation control step does not exist.		Create the specified operation control program.
16202	No specified program (Gn)	• The program (Gn) specified with the transition does not exist.		Create the specified transition program.
16203	No specified program (Motion SFC)	• The Motion SFC program specified with the clear step does not exist.		Correct the specified Motion SFC program name or create the specified Motion SFC program.
16204	No setting of operation expression/conditional expression	• The program (Gn) specified with the transition does not have a conditional expression setting.		Be sure to set a conditional expression in the last block of the transition program.
16205	Fn/FSn program code error	• Internal code error in the operation control program (Fn/FSn)		The Motion SFC program code is corrupted. Turn PLC ready flag (M2000) OFF and write the Motion SFC program again.
16206	Gn program code error	• Internal code error in the transition program (Gn)		Or, replace the external battery if it passed over
16207	Specified the invalid device	• The invalid device (T, C) or shared device outside range in the program is set.		Correct the program which does set the effective device.

12 ERROR CODE LISTS

(4) Operation control/transition execution errors (16300 to 16599)

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
16301	Event task enable (EI) execution error	• Event task enable was executed at except for the normal task.	The block processing on executing is stopped and the next block is executed.	Event task enable may be executed in the normal task only. Correct the program.
16302	Event task disable (DI) execution error	• Event task disable was executed at except for the normal task.		Event task disable may be executed in the normal task only. Correct the program.
16303	Block transfer (BMOV) execution error	• The cam data of the cam No. specified with (D) or (S) is not yet registered to the Motion controller. • The resolution of the cam No. specified with (D) or (S) differs from the number of transferred words specified with (n). • (S) to (S)+(n-1) is outside the device range. • (D) to (D)+(n-1) is outside the device range. • (n) is 0 or a negative number. • (S) is a bit device and the device number is not a multiple of 16. • (D) is a bit device and the device number is not a multiple of 16. • PX/PY is set in (S) to (S)+(n-1). • PX/PY is set in (D) to (D)+(n-1).		• Correct the program so that cam data is that of the already registered cam No.. • Correct the program to match (n) with the cam resolution. • Change (n) so that the block transfer range is within the device range. • Change (n) to a positive number. • When (S) or (D) is a bit device, set the device number to be multiple of 16. • When (S) or (D) is a bit device, do not set PX/PY.
16304	Time to wait (TIME) execution error	• The device No. which indirectly specifies (S) is illegal. • The (S) data is outside the range 0 to 2147483647.		• Correct the program so that the device No. which indirectly specifies (S) is proper. • Correct the program so that the (S) data is within the range of 0 to 2147483647.
16305	Same data block transfer (FMOV) execution error	• (D) to (D)+(n-1) is outside the device range. • (n) is 0 or a negative number. • (S) is a bit device and the device number is not a multiple of 16. • (D) is a bit device and the device number is not a multiple of 16. • PX/PY is set in (S). • PX/PY is set in (D) to (D)+(n-1).		• Change (n) so that the block transfer range is within the device range. • When (S) or (D) is a bit device, set the device number to be multiple of 16. • When (S) or (D) is a bit device, do not set PX/PY.
16308	Speed change request (CHGV) execution error	• The specified axis No. is outside the range.		Correct the program so that the specified axis No. is within the range.
16309	Torque limit value change request (CHGT) execution error			
16316	Assignment (=) execution error	• The (S) data is outside the range of the data type of (D). • The device No. which indirectly specifies (D) is illegal.		• Correct the program so that the (S) data is within the range of the data type of (D). • Correct the program so that the device No. which indirectly specifies (D) is proper.
16320	Operation (/) execution error	• The divisor is 0.		Correct the program so that the divisor is other than 0.
16321	Remainder (%) execution error			
16322	Device set (SET) execution error	• The device No. which indirectly specifies (D) is illegal. • (D) is a device which is write-disabled.		• Correct the program so that the device No. which indirectly specifies (D) is proper. • Correct the program to set a write-enabled device at (D).
16333	Device reset (RST) execution error			
16334	Device set (SET=) execution error			
16335	Device reset (RST=) execution error			
16336	Device output (DOUT) execution error			
16337	Device input (DIN) execution error			
16338	Bit device output (OUT=) execution error			

12 ERROR CODE LISTS

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
16368	Direct specification 16 bit Multiple CPU area device for CPU No.1 (U3E0\G10000 to) read error	<ul style="list-style-type: none"> Multiple CPU area device number is outside the range that set by the parameter. 	The block processing on executing is stopped and the next block is executed.	Correct the program so that Multiple CPU area device number is within the range set in the parameter.
16369	Direct specification 32 bit Multiple CPU area device for CPU No.1 (U3E0\G10000 to) read error			
16370	Direct specification 64 bit Multiple CPU area device for CPU No.1 (U3E0\G10000 to) read error			
16371	Direct specification 16 bit Multiple CPU area device for CPU No.2 (U3E1\G10000 to) read error			
16372	Direct specification 32 bit Multiple CPU area device for CPU No.2 (U3E1\G10000 to) read error			
16373	Direct specification 64 bit Multiple CPU area device for CPU No.2 (U3E1\G10000 to) read error			
16374	Direct specification 16 bit Multiple CPU area device for CPU No.3 (U3E2\G10000 to) read error			
16375	Direct specification 32 bit Multiple CPU area device for CPU No.3 (U3E2\G10000 to) read error			
16376	Direct specification 64 bit Multiple CPU area device for CPU No.3 (U3E2\G10000 to) read error			
16377	Direct specification 16 bit Multiple CPU area device for CPU No.4 (U3E3\G10000 to) read error			
16378	Direct specification 32 bit Multiple CPU area device for CPU No.4 (U3E3\G10000 to) read error			
16379	Direct specification 64 bit Multiple CPU area device for CPU No.4 (U3E3\G10000 to) read error			

12 ERROR CODE LISTS

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
16380	Signed 16-bit integer value conversion (SHORT) execution error	<ul style="list-style-type: none">The (S) data is outside the signed 16-bit integer value range.	The block processing on executing is stopped and the next block is executed.	Correct the program so that the (S) data is within the signed 16-bit integer value range.
16381	Unsigned 16-bit integer value conversion (USHORT) execution error	<ul style="list-style-type: none">The (S) data is outside the unsigned 16-bit integer value range.		Correct the program so that the (S) data is within the unsigned 16-bit integer value range.
16382	Signed 32-bit integer value conversion (LONG) execution error	<ul style="list-style-type: none">The (S) data is outside the signed 32-bit integer value range.		Correct the program so that the (S) data is within the signed 32-bit integer value range.
16383	Unsigned 32-bit integer value conversion (ULONG) execution error	<ul style="list-style-type: none">The (S) data is outside the unsigned 32-bit integer value range.		Correct the program so that the (S) data is within the signed 32-bit integer value range.
16398	Tangent (TAN) execution error	<ul style="list-style-type: none">(S) is $90+(180 \times n)$. (n is an integer)		Correct the program so that (S) is not $90+(180 \times n)$. (n is an integer)
16399	Arcsine (ASIN) execution error	<ul style="list-style-type: none">(S) is outside the range of -1.0 to 1.0.		Correct the program so that (S) is within the range of -1.0 to 1.0.
16400	Arccosine (ACOS) execution error			
16402	Square root (SQRT) execution error	<ul style="list-style-type: none">(S) is a negative number.		Correct the program so that (S) is a positive number.
16403	BCD → BIN conversion (BIN) execution error	<ul style="list-style-type: none">Any digit of (S) has a value other than 0 to 9.		Correct the program so that each digit of (S) is 0 to 9.
16404	BIN → BCD conversion (BCD) execution error	<ul style="list-style-type: none">The (S) value is outside the range where BIN data can be converted into BCD data.		Correct the program so that the (S) value is within the range.
16405	Natural logarithm (LN) execution error	<ul style="list-style-type: none">(S) is 0 or a negative number.		Correct the program so that (S) is a positive number.
16420	Write device data to CPU shared memory of the self CPU (MULTW) execution error	<ul style="list-style-type: none">Number of words (n) to be written is outside the range of 1 to 256.The CPU shared memory address (D) of self CPU of the writing destination device is outside the range (800H to FFFH) of the CPU shared memory address.The CPU shared memory address (D) of self CPU of the writing destination device + number of words (n) to be written is outside the range (800H to FFFH) of the CPU shared memory address.Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range.MULTW instruction was executed again before MULTW instruction is executed and complete bit device is turned on.(D1) is a write-disabled device.(S) is a bit device and the device number is not a multiple of 16.PX/PY is set in (S) to (S)+(n-1).		<ul style="list-style-type: none">Correct the program so that the number of words (n) to be written is within the range of 1 to 256.Correct the program so that the CPU shared memory address (D) of self CPU of the writing destination is within the range of CPU shared memory address.Correct the program so that the CPU shared memory address (D) of self CPU of the writing destination + number of words (n) to be written is within the range of CPU shared memory address.Correct the program so that start device No. (S) which writing data are stored + number of words (n) to be written is within the device range.Execute MULTW instruction again after the complete bit device of MULTW instruction is turned on.Correct the program to set a write-enabled device at (D1).When (S) is a bit device, set the device number to be multiple of 16.When (S) is a bit device, do not set PX/PY.

12 ERROR CODE LISTS

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
16421	Read device data from CPU shared memory of the other CPU (MULTR) execution error	<ul style="list-style-type: none"> Number of words (n) to be read is outside the range of 1 to 256. The CPU shared memory first address (S2) of the data which it will be read is outside the range (000H to FFFH) of the CPU shared memory address. The CPU shared memory first address (S2) of the data which it will be read + number of words (n) to be read is outside the range (000H to FFFH) of the CPU shared memory address. Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range. Except 3E0H/3E1H/3E2H/3E3H is set at (S1). The self CPU is specified with (S1). The CPU which reads is resetting. The errors are detected in the CPU which read. (D) is a bit device and the device number is not a multiple of 16. PX/PY is set in (D) to (D)+(n-1). 		<ul style="list-style-type: none"> Correct the program so that the number of words (n) to be read is within the range of 1 to 256. Correct the program so that the CPU shared memory first address (S2) of the data which it will be read is within the device range of CPU shared memory address. Correct the program so that the CPU shared memory first address (S2) of the data which it will be read + number of words (n) to be read is within the range of CPU shared memory address. Correct the program so that start device No. (D) which stores the reading data + number of words (n) to be read is within the device range. Correct the program so that 3E0H/3E1H/3E2H/3E3H is set at (S1). Correct the program so that the self CPU is not specified with (S1). Check that the resetting flag (SM240 to SM243) is OFF, then correct the program to execute the MULTR instruction. If the errors are detected in the CPU which read, exchange the CPU. When (D) is a bit device, set the device number to be multiple of 16. When (D) is a bit device, do not set PX/PY.
16422	Write device data to intelligent function module (TO) execution error	<ul style="list-style-type: none"> Number of words (n) to be written is outside the range of 1 to 256. Motion CPU cannot communicate with intelligent function module at the instruction execution. Abnormalities of the intelligent function module were detected at the instruction execution. I/O No.s specified with (D1) differ from the intelligent function module controlled by the self CPU. The address specified with (D2) is outside the buffer memory range. Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range. (S) is a bit device and the device number is not a multiple of 16. PX/PY is set in (S) to (S)+(n-1). 	The block processing in execution is stopped and the next block is executed.	<ul style="list-style-type: none"> Correct the program so that the number of words (n) to be written is within the range of 1 to 256. Replace the intelligent function module if there is a fault. Correct the program so that the first I/O No.s specified with (D1) is intelligent function module controlled by the self CPU. Correct the program so that the address specified with (D2) is within the buffer memory range. Correct the program so that start device No. (S) which writing data are stored + number of words (n) to be written is within the device range. When (S) is a bit device, set the device number to be multiple of 16. When (S) is a bit device, do not set PX/PY.
16423	Read device data from intelligent function module (FROM) execution error	<ul style="list-style-type: none"> Number of words (n) to be read is outside the range of 1 to 256. Motion CPU cannot communicate with intelligent function module at the instruction execution. Abnormalities of the intelligent function module were detected at the instruction execution. I/O No.s specified with (S1) differ from the intelligent function module controlled by the self CPU. The address specified with (S2) is outside the range buffer memory. Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range. (D) is a bit device and the device number is not a multiple of 16. PX/PY is set in (D) to (D)+(n-1). 		<ul style="list-style-type: none"> Correct the program so that the number of words (n) to be read is within the range of 1 to 256. Replace the intelligent function module if there is a fault. Correct the program so that the first I/O No.s specified with (S1) is intelligent function module controlled by the self CPU. Correct the program so that the address specified with (S2) is within the buffer memory range. Correct the program so that start device No. (D) which stores the reading data + number of words(n) to be read is within the device range. When (D) is a bit device, set the device number to be multiple of 16. When (D) is a bit device, do not set PX/PY.

12 ERROR CODE LISTS

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
16441	Indirect specified 16-bit SD(SD(n)) read error	• The indirectly specified device No. is outside the range.	The block processing in execution is stopped and the next block is executed.	Correct the program so that the indirectly specified device No. is proper.
16442	Indirect specified 32-bit SD(SD(n)L) read error	• The indirectly specified device No. is outside the range or an odd number.		
16443	Indirect specified 64-bit SD(SD(n)F) read error			
16462	Indirect specified 32-bit motion device (#(n)) read error	• The indirectly specified device No. is outside the range.		
16463	Indirect specified 64-bit motion device (#(n)L) read error	• The indirectly specified device No. is outside the range or an odd number.		
16464	Indirect specified 64-bit motion device (#(n)F) read error			
16465	Indirect specified 16-bit data register (D(n)) read error	• The indirectly specified device No. is outside the range.		
16466	Indirect specified 32-bit data register (D(n)L) read error	• The indirectly specified device No. is outside the range or an odd number.		
16467	Indirect specified 64-bit data register (D(n)F) read error			
16468	Indirect specified 16-bit link register (W(n)) read error	• The indirectly specified device No. is outside the range.		
16469	Indirect specified 32-bit link register (W(n)L) read error	• The indirectly specified device No. is outside the range or an odd number.		
16470	Indirect specified 64-bit link register (W(n)F) read error			
16475	Indirect specified SM(SM(n)) read error	• The indirectly specified device No. is outside the range.		
16482	Direct specified Multiple CPU area device bit specified for CPU No.1 (U3E0\G10000.0 to) read error	• Multiple CPU area device number is outside the range set in the parameter.		Correct the program so that Multiple CPU area device number is within the range set in the parameter.
16483	Direct specified Multiple CPU area device bit specified for CPU No.2 (U3E1\G10000.0 to) read error			
16484	Direct specified Multiple CPU area device bit specified for CPU No.3 (U3E2\G10000.0 to) read error			
16485	Direct specified Multiple CPU area device bit specified for CPU No.4 (U3E3\G10000.0 to) read error			

12 ERROR CODE LISTS

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
16486	Indirect specified input relay (X(n)) read error	• The indirectly specified device No. is outside the range.	The block processing in execution is stopped and the next block is executed.	Correct the program so that the indirectly specified device No. is proper.
16487	Indirect specified output relay (Y(n)) read error			
16488	Indirect specified internal relay (M(n)) read error			
16489	Indirect specified link relay (B(n)) read error			
16490	Annunciator (F(n)) read error			
16516	Indirect specified 16-bit batch input relay (X(n)) read error	• The indirectly specified device No. is outside the range or is not a multiple of 16.		
16517	Indirect specified 32-bit batch input relay (X(n)) read error			
16518	Indirect specified 16-bit batch output relay (Y(n)) read error			
16519	Indirect specified 32-bit batch output relay (Y(n)) read error			
16520	Indirect specified 16-bit batch internal relay (M(n)) read error			
16521	Indirect specified 32-bit batch internal relay (M(n)) read error			
16522	Indirect specified 16-bit batch internal/latch relay (B(n)) read error			
16523	Indirect specified 32-bit batch internal/latch relay (B(n)) read error			
16524	Indirect specified 16-bit batch annunciator (F(n)) read error			
16525	Indirect specified 32-bit batch annunciator (F(n)) read error			
16538	Indirect specified 16-bit batch SM(SM(n)) read error			
16539	Indirect specified 32-bit batch SM(SM(n)) read error			

12 ERROR CODE LISTS

12.4 Motion SFC Parameter Errors

Motion SFC parameters are checked using MT Developer.

(1) Leading edge of PLC ready flag (M2000) errors (17000 to 17009)

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
17000	Normal task consecutive transition count error	• The normal task's consecutive transition count of the Motion SFC program started by the normal task is outside the range 1 to 30.	The initial value of 3 is used for control.	Turn PLC ready flag (M2000) OFF, make correction to set the value within the range, and write it to the CPU.
17001	Event task consecutive transition count error	• The set number of consecutive transitions of the Motion SFC program started by the event task is outside the range 1 to 10.	The initial value of 1 is used for control.	
17002	NMI task consecutive transition count error	• The set number of consecutive transitions of the Motion SFC program started by the NMI task is outside the range 1 to 10.		
17003	Motion SFC parameter unregistered error	• Motion SFC parameter is not written or parameter is corrupted.	The initial value of Motion SFC parameter is used for control.	Turn PLC ready flag (M2000) OFF and write the Motion SFC parameter.

(2) SFC Program start errors (17010 to 17019)

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
17010	Executed task setting is illegal	• Among the normal, event and NMI tasks, more than one or none of them has been set.	The initial value (normal task) is used for control.	Turn PLC ready flag (M2000) OFF, make correction, and write a correct value to the CPU.
17011	Executed task setting is illegal (event)	• Two or more fixed cycles of the event task have been set.		

APPENDICES

APPENDIX 1 Processing Times

APPENDIX 1.1 Processing time of operation control/Transition instruction

(1) Operation instructions

Processing time of operation instructions

Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μs]
Binary operation	=	Substitution	#0=#1	1.11
			D800=D801	1.64
			U3E1\G10000=U3E1\G10001	1.97
			#0L=#2L	1.25
			D800L=D802L	1.64
			U3E1\G10000L=U3E1\G10002L	1.91
			#0F=#4F	1.60
			D800F=D804F	1.78
			U3E1\G10000F=U3E1\G10004F	2.67
	+	Addition	#0=#1+#2	1.76
			D800=D801+D802	2.22
			U3E1\G10000=U3E1\G10001+U3E1\G10002	3.00
			#0L=#2L+#4L	1.87
			D800L=D802L+D804L	2.34
			U3E1\G10000L=U3E1\G10002L+U3E1\G10004L	3.14
			#0F=#4F+#8F	2.44
			D800F=D804F+D808F	2.82
			U3E1\G10000F=U3E1\G10004F+U3E1\G10008F	4.41
	-	Subtraction	#0=#1-#2	2.25
			D800=D801-D802	2.37
			U3E1\G10000=U3E1\G10001-U3E1\G10002	3.34
			#0L=#2L-#4L	2.22
			D800L=D802L-D804L	2.90
			U3E1\G10000L=U3E1\G10002L-U3E1\G10004L	3.24
			#0F=#4F-#8F	2.77
			D800F=D804F-D808F	3.14
			U3E1\G10000F=U3E1\G10004F-U3E1\G10008F	4.78
	*	Multiplication	#0=#1*#2	2.46
			D800=D801*D802	2.49
			U3E1\G10000=U3E1\G10001*U3E1\G10002	3.68
			#0L=#2L*#4L	2.42
			D800L=D802L*D804L	2.71
			U3E1\G10000L=U3E1\G10002L*U3E1\G10004L	3.66
			#0F=#4F*#8F	2.70
			D800F=D804F*D808F	3.63
			U3E1\G10000F=U3E1\G10004F*U3E1\G10008F	4.87
	/	Division	#0=#1/#2	2.30
			D800=D801/D802	2.46
			U3E1\G10000=U3E1\G10001/U3E1\G10002	3.39

Processing time of operation instructions (Continued)

Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μs]
Binary operation	/	Division	#0L=#2L/#4L	2.19
			D800L=D802L/D804L	2.67
			U3E1\G10000L=U3E1\G10002L/U3E1\G10004L	3.35
			#0F=#4F/#8F	2.94
			D800F=D804F/D808F	3.54
			U3E1\G10000F=U3E1\G10004F/U3E1\G10008F	4.58
	%	Remainder	#0=#1%#2	2.07
			D800=D801%D802	2.48
			U3E1\G10000=U3E1\G10001%U3E1\G10002	3.06
			#0L=#2L%#4L	2.01
			D800L=D802L%D804L	2.55
			U3E1\G10000L=U3E1\G10002L%U3E1\G10004L	3.25
Bit operation	~	Bit inversion (complement)	#0=~#1	1.22
			D800=~D801	1.59
			U3E1\G10000=~U3E1\G10001	1.98
			#0L=~#2L	1.25
			D800L=~D802L	1.63
			U3E1\G10000L=~U3E1\G10002L	2.29
	&	Bit logical AND	#0=#1	2.00
			D800=D801&D802	2.27
			U3E1\G10000=U3E1\G10001&U3E1\G10002	3.11
			#0L=#2LL	1.75
			D800L=D802L&D804L	2.30
			U3E1\G10000L=U3E1\G10002L&U3E1\G10004L	3.17
		Bit logical OR	#0=#1 #2	1.81
			D800=D801 D802	2.02
			U3E1\G10000=U3E1\G10001 U3E1\G10002	2.83
			#0L=#2L #4L	1.74
			D800L=D802L D804L	2.32
			U3E1\G10000L=U3E1\G10002L U3E1\G10004L	2.94
	^	Bit exclusive OR	#0=#1^#2	1.74
			D800=D801^D802	2.29
			U3E1\G10000=U3E1\G10001^U3E1\G10002	2.85
			#0L=#2L^#4L	1.78
			D800L=D802L^D804L	2.39
			U3E1\G10000L=U3E1\G10002L^U3E1\G10004L	3.03
	>>	Bit right shift	#0=#1>>#2	2.13
			D800=D801>>D802	2.31
			U3E1\G10000=U3E1\G10001>>U3E1\G10002	3.30
			#0L=#2L>>#4L	2.04
			D800L=D802L>>D804L	2.44
			U3E1\G10000L=U3E1\G10002L>>U3E1\G10004L	3.08
	<<	Bit left shift	#0=#1<<#2	1.81
			D800=D801<<D802	2.31
			U3E1\G10000=U3E1\G10001<<U3E1\G10002	3.21
			#0L=#2L<<#4L	1.89
			D800L=D802L<<D804L	2.30
			U3E1\G10000L=U3E1\G10002L<<U3E1\G10004L	3.00

Processing time of operation instructions (Continued)

Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μ s]
Sign	-	Sign inversion (complement of 2)	#0=#1	1.12
			D800=-D812	1.62
			U3E1\G10000=-U3E1\G10001	2.13
			#0L=#2L	1.44
			D800L=-D802L	1.94
			U3E1\G10000L=-U3E1\G10002L	2.19
			#0F=#4F	1.86
			D800F=-D804F	2.28
			U3E1\G10000F=-U3E1\G10004F	2.94
Standard function	SIN	Sine	#0F=SIN(#4F)	5.13
			D800F=SIN(D804F)	4.00
			U3E1\G10000F=SIN(U3E1\G10004F)	5.48
	COS	Cosine	#0F=COS(#4F)	4.31
			D800F=COS(D804F)	4.41
			U3E1\G10000F=COS(U3E1\G10004F)	5.46
	TAN	Tangent	#0F=TAN(#4F)	5.80
			D800F=TAN(D804F)	5.48
			U3E1\G10000F=TAN(U3E1\G10004F)	6.70
	ASIN	Arcsine	#0F=ASIN(#4F)	13.49
			D800F=ASIN(D804F)	11.27
			U3E1\G10000F=ASIN(U3E1\G10004F)	14.37
	ACOS	Arccosine	#0F=ACOS(#4F)	10.11
			D800F=ACOS(D804F)	10.18
			U3E1\G10000F=ACOS(U3E1\G10004F)	11.24
	ATAN	Arctangent	#0F=ATAN(#4F)	4.44
			D800F=ATAN(D804F)	4.49
			U3E1\G10000F=ATAN(U3E1\G10004F)	5.64
	SQRT	Square root	#0F=SQRT(#4F)	2.13
			D800F=SQRT(D804F)	2.42
			U3E1\G10000F=SQRT(U3E1\G10004F)	3.18
	LN	Natural logarithm	#0F=LN(#4F)	5.75
			D800F=LN(D804F)	4.68
			U3E1\G10000F=LN(U3E1\G10004F)	5.49
	EXP	Exponential operation	#0F=EXP(#4F)	4.06
			D800F=EXP(D804F)	3.65
			U3E1\G10000F=EXP(U3E1\G10004F)	4.59
	ABS	Absolute value	#0F=ABS(#4F)	1.74
			D800F=ABS(D804F)	2.03
			U3E1\G10000F=ABS(U3E1\G10004F)	3.03
	RND	Round-off	#0F=RND(#4F)	2.55
			D800F=RND(D804F)	2.49
			U3E1\G10000F=RND(U3E1\G10004F)	3.57
	FIX	Round-down	#0F=FIX(#4F)	2.18
			D800F=FIX(D804F)	2.43
			U3E1\G10000F=FIX(U3E1\G10004F)	3.18
	FUP	Round-up	#0F=FUP(#4F)	2.49
			D800F=FUP(D804F)	2.42
			U3E1\G10000F=FUP(U3E1\G10004F)	3.42

Processing time of operation instructions (Continued)

Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μ s]
Standard function	BIN	BCD→BIN conversion	#0=BIN(#1)	1.65
			D800=BIN(D801)	1.92
			U3E1\G10000=BIN(U3E1\G10001)	2.43
			#0L=BIN(#2L)	2.21
			D800L=BIN(D802L)	2.34
			U3E1\G10000L=BIN(U3E1\G10002L)	2.66
	BCD	BIN→BCD conversion	#0=BCD(#1)	1.89
			D800=BCD(D801)	2.26
			U3E1\G10000=BCD(U3E1\G10001)	2.73
			#0L=BCD(#2L)	2.44
			D800L=BCD(D802L)	2.58
			U3E1\G10000L=BCD(U3E1\G10002L)	3.17
Type conversion	SHORT	Converted into 16-bit integer type (signed)	#0=SHORT(#2L)	1.64
			#0=SHORT(#4F)	2.19
			D800=SHORT(D802L)	1.89
			D800=SHORT(D804F)	2.46
			U3E1\G10000=SHORT(U3E1\G10002L)	2.33
			U3E1\G10000=SHORT(U3E1\G10004F)	3.26
	USHORT	Converted into 16-bit integer type (unsigned)	#0=USHORT(#2L)	1.61
			#0=USHORT(#4F)	2.37
			D800=USHORT(D802L)	2.10
			D800=USHORT(D804F)	2.49
			U3E1\G10000=USHORT(U3E1\G10002L)	2.41
			U3E1\G10000=USHORT(U3E1\G10004F)	3.33
	LONG	Converted into 32-bit integer type (signed)	#0L=LONG(#2)	1.47
			#0L=LONG(#4F)	2.59
			D800L=LONG(D802)	1.85
			D800L=LONG(D804F)	2.62
			U3E1\G10000L=LONG(U3E1\G10002L)	2.22
			U3E1\G10000L=LONG(U3E1\G10004F)	3.45
	ULONG	Converted into 32-bit integer type (unsigned)	#0L=ULONG(#2)	1.65
			#0L=ULONG(#4F)	2.94
			D800L=ULONG(D802)	2.01
			D800L=ULONG(D804F)	2.98
			U3E1\G10000L=ULONG(U3E1\G10002L)	2.41
			U3E1\G10000L=ULONG(U3E1\G10004F)	3.88
	FLOAT	Converted into 64-bit floating point type (signed)	#0F=FLOAT(#4)	1.44
			#0F=FLOAT(#4L)	1.34
			D800F=FLOAT(D804)	1.85
			D800F=FLOAT(D804L)	2.30
			U3E1\G10000F=FLOAT(U3E1\G10004)	2.28
			U3E1\G10000F=FLOAT(U3E1\G10004L)	2.92
	UFLOAT	Converted into 64-bit floating point type (unsigned)	#0F=UFLOAT(#4)	1.39
			#0F=UFLOAT(#4L)	1.69
			D800F=UFLOAT(D804)	1.87
			D800F=UFLOAT(D804L)	1.77
			U3E1\G10000F=UFLOAT(U3E1\G10004)	2.31
			U3E1\G10000F=UFLOAT(U3E1\G10004L)	2.17

Processing time of operation instructions (Continued)

Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μ s]
Bit device status	(None)	ON (normally open contact) (Completion of condition)	SET M1000 = M0	2.54
			SET M1000 = X100	3.12
			SET M1000 = PX0	4.85
			SET M1000 = U3E1\G10000.0	3.30
	!	OFF (normally closed contact) (Completion of condition)	SET M1000 = !M0	2.80
			SET M1000 = !X100	2.98
			SET M1000 = !PX0	4.48
			SET M1000 = !U3E1\G10000.0	4.86
Bit device control	SET	Device set	SET M1000	1.78
			SET Y100	2.59
			SET PY0	1.86
			SET U3E1\G11000.0	2.14
	RST	Device reset	RST M1000	1.74
			RST Y100	2.50
			RST PY0	1.81
			RST U3E1\G11000.0	2.63
	DOUT	Device output	DOUT M0,#0	2.70
			DOUT M0,#0L	2.46
			DOUT Y100,#0	2.33
			DOUT Y100,#0L	4.00
			DOUT PY0,#0	2.27
			DOUT PY0,#0L	3.54
	DIN	Device input	DIN #0,M0	2.81
			DIN #0L,M0	2.65
			DIN #0,X0	2.15
			DIN #0L,X0	2.76
			DIN #0,PX0	4.32
			DIN #0L,PX0	8.96
	OUT	Bit device output	OUT M1000 = M0	2.28
			OUT Y0 = M0	2.76
			OUT PY0 = M0	2.49
			OUT U3E1\G10000.0 = M0	3.45
Logical operation	*	Logical AND	SET M1000 = M0*M1	3.07
			SET M1000 = X100*X101	3.61
			SET M1000 = PX0*PX1	6.18
			SET M1000 = U3E1\G10000.0*U3E1\G10000.1	3.19
	+	Logical OR	SET M1000 = M0+M1	3.01
			SET M1000 = X100+X101	3.56
			SET M1000 = PX0+PX1	8.92
			SET M1000 = U3E1\G10000.0+U3E1\G10000.1	3.36

Processing time of operation instructions (Continued)

Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μ s]
Comparison operation	==	Equal to (Completion of condition)	SET M1000 = #0=#1	3.38
			SET M1000 = D800==D801	3.79
			SET M1000 = U3E1\G10000==U3E1\G10001	4.41
			SET M1000 = #0L==#2L	3.46
			SET M1000 = D800L==D802L	3.91
			SET M1000 = U3E1\G10000L==U3E1\G10002L	4.49
			SET M1000 = #0F==#4F	4.17
			SET M1000 = D800F==D804F	4.48
			SET M1000 = U3E1\G10000F==U3E1\G10004F	5.65
	!=	Not equal to (Completion of condition)	SET M1000 = #0!=#1	3.52
			SET M1000 = D800!=D801	3.80
			SET M1000 = U3E1\G10000!=U3E1\G10001	4.35
			SET M1000 = #0L!=#2L	3.52
			SET M1000 = D800L!=D802L	3.73
			SET M1000 = U3E1\G10000L!=U3E1\G10002L	4.45
			SET M1000 = #0F!=#4F	4.04
			SET M1000 = D800F!=D804F	4.63
			SET M1000 = U3E1\G10000F!=U3E1\G10004F	5.87
	<	Less than (Completion of condition)	SET M1000 = #0<#1	3.47
			SET M1000 = D800<D801	3.85
			SET M1000 = U3E1\G10000<U3E1\G10001	4.36
			SET M1000 = #0L<#2L	3.55
			SET M1000 = D800L<D802L	3.95
			SET M1000 = U3E1\G10000L<U3E1\G10002L	4.47
			SET M1000 = #0F<#4F	3.97
			SET M1000 = D800F<D804F	4.48
			SET M1000 = U3E1\G10000F<U3E1\G10004F	5.84
	<=	Less than or equal to (Completion of condition)	SET M1000 = #0<=#1	3.39
			SET M1000 = D800<=D801	3.76
			SET M1000 = U3E1\G10000<=U3E1\G10001	4.46
			SET M1000 = #0L<=#2L	3.46
			SET M1000 = D800L<=D802L	3.76
			SET M1000 = U3E1\G10000L<=U3E1\G10002L	4.59
			SET M1000 = #0F<=#4F	4.10
			SET M1000 = D800F<=D804F	4.48
			SET M1000 = U3E1\G10000F<=U3E1\G10004F	5.70
	>	More than (Completion of condition)	SET M1000 = #0>#1	3.76
			SET M1000 = D800>D801	3.71
			SET M1000 = U3E1\G10000>U3E1\G10001	4.23
			SET M1000 = #0L>#2L	3.54
			SET M1000 = D800L>D802L	3.90
			SET M1000 = U3E1\G10000L>U3E1\G10002L	4.46
			SET M1000 = #0F>#4F	4.00
			SET M1000 = D800F>D804F	4.46
			SET M1000 = U3E1\G10000F>U3E1\G10004F	5.78

Processing time of operation instructions (Continued)

Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μ s]
Comparison operation	>=	More than or equal to (Completion of condition)	SET M1000 = #0>=#1	3.66
			SET M1000 = D800>=D801	3.59
			SET M1000 = U3E1\G10000>=U3E1\G10001	4.42
			SET M1000 = #0L>=#2L	3.58
			SET M1000 = D800L>=D802L	4.20
			SET M1000 = U3E1\G10000L>=U3E1\G10002L	4.61
			SET M1000 = #0F>=#4F	4.06
			SET M1000 = D800F>=D804F	4.73
			SET M1000 = U3E1\G10000F>=U3E1\G10004F	5.89
Motion dedicated function	CHGV	Speed change request	CHGV(K1,#0)	5.65
			CHGV(K1,D800)	4.62
			CHGV(K1,U3E1\G10000)	4.41
			CHGV(K1,#0L)	4.34
			CHGV(K1,D800L)	4.84
			CHGV(K1,U3E1\G10000L)	3.48
	CHGT	Torque limit value change request	CHGT(K1,#0)	3.16
			CHGT(K1,D800)	3.44
			CHGT(K1,U3E1\G10000)	2.40
			CHGT(K1,#0L)	3.47
			CHGT(K1,D800L)	3.88
			CHGT(K1,U3E1\G10000L)	2.66
Others	EI	Event task enable	EI	0.31
	DI	Event task disable	DI	0.43
	NOP	No operation	NOP	0.32
	BMOV	Block transfer	BMOV #0,#100,K10	5.88
			BMOV D800,D100,K10	5.05
			BMOV U3E1\G10000,U3E1\G10100,K10	7.26
			BMOV #0,#100,K100	13.98
			BMOV D800,D100,K100	23.76
			BMOV U3E1\G10000,U3E1\G10100,K100	5.63
			BMOV N1,#0,K512	123.90
			BMOV N1,D800,K512	122.48
			BMOV N1,U3E1\G10000,K512	250.47
	FMOV	Same data block transfer	FMOV #0,#100,K10	3.62
			FMOV D800,D100,K10	3.29
			FMOV U3E1\G10000,U3E1\G10100,K10	3.67
			FMOV #0,#100,K100	5.56
			FMOV D800,D100,K100	9.08
			FMOV 3E1\G10000,U3E1\G10100,K100	3.39
	MULTW	Write device data to CPU shared memory of the self CPU	MULTW H800,#0,K1,M0	3.87
			MULTW H800,D800,K1,M0	3.55
			MULTW H800,U3E1\G10000,K1,M0	4.98
			MULTW H800,#0,K10,M0	5.42
			MULTW H800,D800,K10,M0	5.41
			MULTW H800,U3E1\G10000,K10,M0	9.32
			MULTW H800,#0,K100,M0	23.46
			MULTW H800,D800,K100,M0	23.34
			MULTW H800,U3E1\G10000,K100,M0	61.08

Processing time of operation instructions (Continued)

Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μs]
Others	MULTW	Write device data to CPU shared memory of the self CPU	MULTW H800,#0,K256,M0	57.80
			MULTW H800,D800,K256,M0	57.69
			MULTW H800,U3E1\G10000,K256,M0	151.28
	MULTR	Read device data from CPU shared memory of the other CPU	MULTR #0,H3E0,H800,K1	21.00
			MULTR D800,H3E0,H800,K1	20.10
			MULTR U3E1\G10000,H3E0,H800,K1	21.82
			MULTR #0,H3E0,H800,K10	30.20
			MULTR D800,H3E0,H800,K10	30.07
			MULTR U3E1\G10000,H3E0,H800,K10	31.14
			MULTR #0,H3E0,H800,K100	140.74
			MULTR D800,H3E0,H800,K100	139.99
			MULTR U3E1\G10000,H3E0,H800,K100	152.01
			MULTR #0,H3E0,H800,K256	411.57
			MULTR D800,H3E0,H800,K256	412.01
			MULTR U3E1\G10000,H3E0,H800,K256	434.79
	TO	Write device data to intelligent function module	TO H0,H0,#0,K1	15.71
			TO H0,H0,D800,K1	14.61
			TO H0,H0,U3E1\G10000,K1	16.07
			TO H0,H0,#0,K10	18.66
			TO H0,H0,D800,K10	18.07
			TO H0,H0,U3E1\G10000,K10	21.60
			TO H0,H0,#0,K100	84.24
			TO H0,H0,D800,K100	83.65
			TO H0,H0,U3E1\G10000,K100	121.32
			TO H0,H0,#0,K256	262.31
			TO H0,H0,D800,K256	185.07
			TO H0,H0,U3E1\G10000,K256	358.23
	FROM	Read device data from intelligent function module	FROM #0,H0,H0,K1	13.27
			FROM D800,H0,D800,K1	15.24
			FROM U3E1\G10000,H0,U3E1\G10000,K1	16.21
			FROM #0,H0,#0,K10	22.34
			FROM D800,H0,H0,K10	23.97
			FROM U3E1\G10000,H0,U3E1\G10000,K10	22.89
			FROM #0,H0,#0,K100	132.52
			FROM D800,H0,H0,K100	131.56
			FROM U3E1\G10000,H0,U3E1\G10000,K100	144.43
			FROM #0,H0,H0,K256	407.29
			FROM D800,H0,H0,K256	402.30
			FROM U3E1\G10000,H0,U3E1\G10000,K256	432.06
	TIME	Time to wait	TIME K1	2.32
			TIME #0	2.38
			TIME D800	2.71
			TIME U3E1\G10000	3.10

(2) Transition conditional expressions

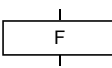
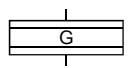
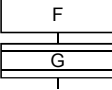
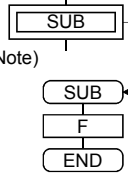
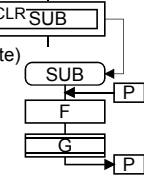

Processing time of transition condition expressions

Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μs]
Bit device control	(None)	ON (Normally open contact) (Completion of condition)	M0	1.04
			X100	1.02
			PX0	2.83
			U3E1\G10000.0	1.28
	!	OFF (Normally closed contact) (Completion of condition)	!M0	1.10
			!X100	1.30
			!PX0	3.18
			!U3E1\G10000.0	1.32
Logical operation	*	Logical AND	M0*M1	1.75
			X100*X101	1.98
			PX0*PX1	5.35
			U3E1\G10000.0*U3E1\G10000.1	2.38
	+	Logical OR	M0+M1	1.59
			X100+X101	1.93
			PX0+PX1	6.90
			U3E1\G10000.0+U3E1\G10000.1	2.32
Comparison operation	==	Equal to (Completion of condition)	#0==#1	1.15
			D800==D801	1.88
			U3E1\G10000==U3E1\G10001	2.31
			#0L==#2L	1.24
			D800L==D802L	2.14
			U3E1\G10000L==U3E1\G10002L	2.51
			#0F==#4F	1.93
			D800F==D804F	2.62
			U3E1\G10000F==U3E1\G10004F	3.86
	!=	Not equal to (Completion of condition)	#0!=#1	1.18
			D800!=D801	1.64
			U3E1\G10000!=U3E1\G10001	2.28
			#0L!=#2L	1.21
			D800L!=D802L	1.90
			U3E1\G10000L!=U3E1\G10002L	2.47
			#0F!=#4F	1.80
			D800F!=D804F	2.25
			U3E1\G10000F!=U3E1\G10004F	3.57
	<	Less than (Completion of condition)	#0<#1	1.33
			D800<D801	1.66
			U3E1\G10000<U3E1\G10001	2.44
			#0L<#2L	1.43
			D800L<D802L	2.03
			U3E1\G10000L<U3E1\G10002L	2.50
			#0F<#4F	1.78
			D800F<D804F	2.31
			U3E1\G10000F<U3E1\G10004F	3.55

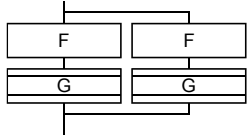
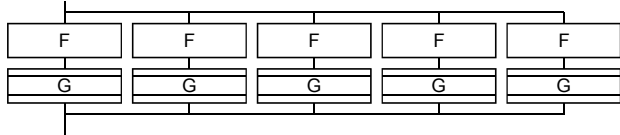
Processing time of operation instructions (Continued)

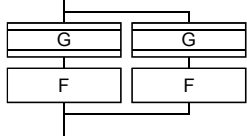
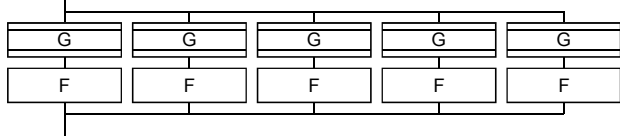
Classifications	Symbol	Instruction	Operation expression	Q173DCPU/Q172DCPU Unit [μs]
Comparison operation	<=	Less than or equal to (Completion of condition)	#0<=#1	1.18
			D800<=D801	1.76
			U3E1\G10000<=U3E1\G10001	2.31
			#0L<=#2L	1.39
			D800L<=D802L	1.75
			U3E1\G10000L<=U3E1\G10002L	2.53
			#0F<=#4F	1.92
			D800<=D804F	2.28
			U3E1\G10000F<=U3E1\G10004F	3.55
	>	More than (Completion of condition)	#0>#1	1.17
			D800>D801	1.77
			U3E1\G10000>U3E1\G10001	2.41
			#0L>#2L	1.23
			D800L>D802L	1.90
			U3E1\G10000L>U3E1\G10002L	2.48
			#0F>#4F	1.79
			D800F>D804F	2.11
			U3E1\G10000F>U3E1\G10004F	3.54
	>=	More than or equal to (Completion of condition)	#0>=#1	1.31
			D800>=D801	1.89
			U3E1\G10000>=U3E1\G10001	2.43
			#0L>=#2L	1.26
			D800L>=D802L	1.87
			U3E1\G10000L>=U3E1\G10002L	2.47
			#0F>=#4F	1.76
			D800F>=D804F	2.34
			U3E1\G10000F>=U3E1\G10004F	3.65

(3) Processing time by the combination F and G (program described in F/G is NOP)

	F alone	G alone	F+G	G SUB	CLR	JMP/coupling
						
Q173DCPU/ Q172DCPU [μs]	13.99	13.18	15.47	22.07	14.54	4.44

(Note): Varies greatly with the started or cleared program.

	Parallel branch (2 Pcs.)		Parallel branch (5 Pcs.)	
				
	At branch	At coupling	At branch	At coupling
Q173DCPU/ Q172DCPU [μs]	22.89	18.51	49.09	32.11

	Selective branch (2 Pcs.)	Selective branch (5 Pcs.)
		
Q173DCPU/ Q172DCPU [μs]	47.97	54.81

POINT

Long processing time may cause a Motion CPU WDT error or servo fault.
Especially for the Motion SFC programs run by event/NMI tasks, take care so that the processing time will not be too long.

APPENDIX 1.2 Processing time of Motion dedicated PLC instruction

Processing time of Motion dedicated PLC instruction

Classifications	Symbol	Instruction (Condition)	Processing time [μs]		
			Q03UDCPU	Q04UDHCPU	Q06UDHCPU
Multiple CPU high speed bus Motion dedicated instruction	D.SFCS	Start request of the specified Motion SFC program	131	131	
	D.SVST	Start request of the specified servo program	147	147	
	D.CHGA	Current value change request of the specified axis	147	147	
	D.CHGV	Speed change request of the specified axis	149	148	
	D.CHGT	Torque control value change request of the specified axis	146	146	
Multiple CPU high speed bus other CPU access instruction	D.DDWR	Write device data of the self CPU to the device of other CPU	Number of writing data = 1	163	160
			Number of writing data = 16	172	167
	D.DDRD	Read device data of other CPU to the device of self CPU	Number of reading data = 1	161	158
			Number of reading data = 16	161	157
	D.GINT	Execute request of an event task of Motion SFC program	132	131	

APPENDIX 2 Sample Program

APPENDIX 2.1 Motion control example by Motion SFC program

- (1) The Motion SFC program composition example to execute motion control.

This sample program example is described to every following function.

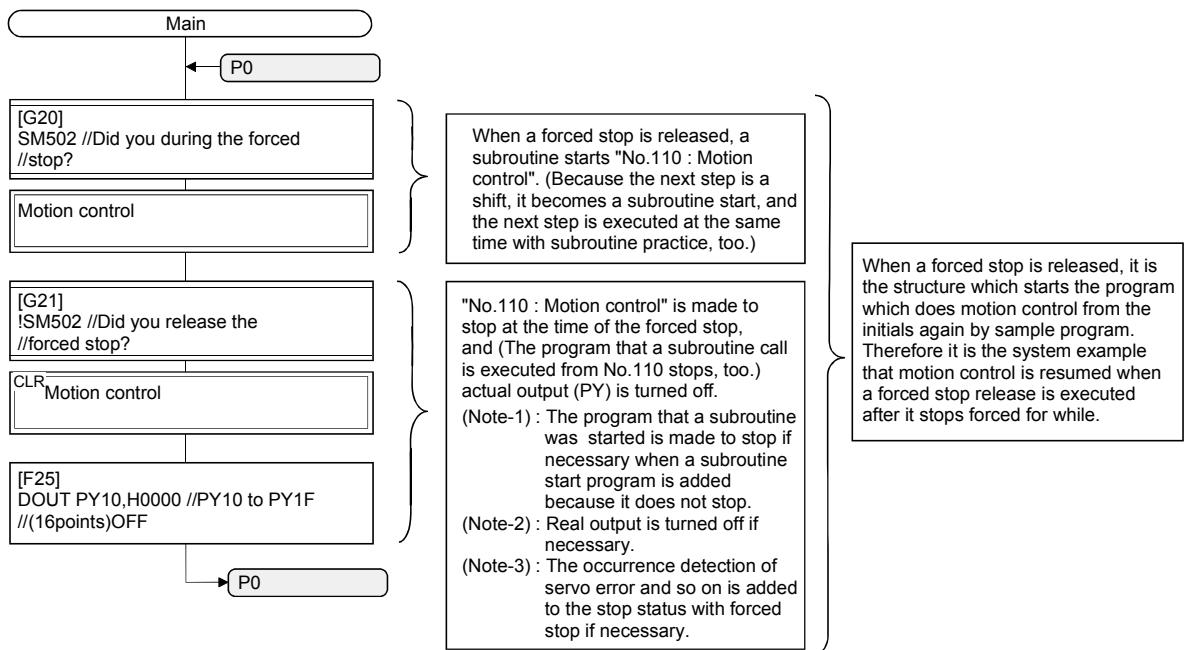
Function list of sample program

No.	Item	Description
1	Forced stop	When the forced stop input assigned to PX0 is on, all axes turn on, and motion control is executed. When the forced stop input turn off, servo amplifier is made to forced stop, and motion control is suspended, and actual output (PY) turn off.
2	Motion control	Motion control is executed according to the condition of PX and PX2 in each following mode. <ul style="list-style-type: none"> • PX2 : OFF PX1 : OFF JOG mode • PX2 : OFF PX1 : ON Manual pulse generator mode • PX2 : On PX1 : OFF Home position return mode • PX2 : On PX1 : On Programming operation mode
3	JOG mode	The following JOG operation is executed when each signal of PX3 to PX6 is turned on. <ul style="list-style-type: none"> • PX3 : 1 axis JOG forward rotation • PX4 : 1 axis JOG reverse rotation • PX5 : 2 axes JOG forward rotation • PX6 : 2 axes JOG reverse rotation
4	Manual pulse generator mode	The following the manual pulse generator operation is executed. <ul style="list-style-type: none"> • Manual pulse generator operation of 1 axis is executed with the manual pulse generator P1. • Manual pulse generator operation of 2 axes is executed with the manual pulse generator P1.
5	Home position return mode	The following home position return is executed. <ul style="list-style-type: none"> • When PX3 is on, the home position return of 1 axis is executed. • When PX4 is on, the home position return of 2 axes is executed.
6	Programming operation mode	The following program operation is executed. <ul style="list-style-type: none"> • When PX3 detects OFF to ON, axis No.1 locates and 1000[ms] standing by, after the location of axis No.2 is executed. • When PX4 turn on, axis No.1, 2 locates of the linear control and in-position check is executed, after the location of axis No.2 is executed, the program stands by until No.1, 2 locates of the linear control is executed at a double speed in the opposition direction and PX4 turns off.

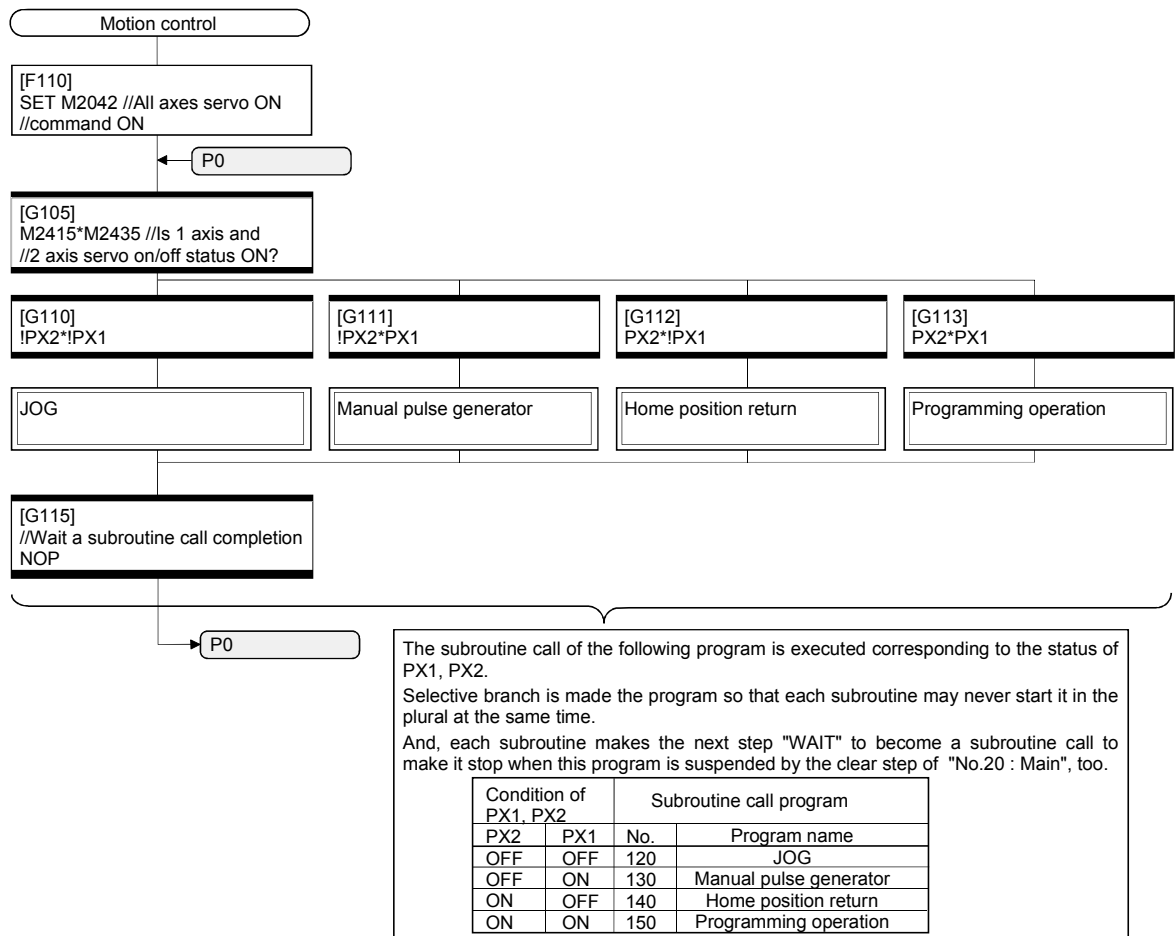
(2) Contents processing of the Motion SFC program
Motion SFC program list

No.	Program name	Task	Automatic operation	Number of connective transitions	Contents of processing
20	Main	Normal	Start	3	<ul style="list-style-type: none"> • This program starts automatically at the time of run of Q173DCPU, and it is always executed. • When a forced stop is canceled, a subroutine starts a "No.110 : Motion control". • "No.110 : Motion control" is stopped at the time of the forced stop, and real output (PY) is turned off.
110	Motion control	Normal	Not start	3	<ul style="list-style-type: none"> • All axes servo on. • The call of the subroutine of the following program is executed by the condition of PX1, PX2. <ul style="list-style-type: none"> 1) PX2 : OFF PX1 : OFF No.120 : JOG 2) PX2 : OFF PX1 : ON No.130 : Manual pulse generator 3) PX2 : ON PX1 : OFF No.140 : Home position return 4) PX2 : ON PX1 : ON No.150 : Programming operation
120	JOG	Normal	Not start	3	<ul style="list-style-type: none"> (1) The JOG operation speed of 1 axis and 2 axes is set. (2) 1 axis JOG forward command is turned on when PX3 is on, and the reverse command is turned on when PX4 is on. (3) 2 axes JOG forward command is turned on when PX5 is on, and the reverse command is turned on when PX6 is on. (4) The above (2), (3) are repeated during PX2/PX1 is off, when except for it, the JOG forward and reverse command of 1 axis and 2 axes are turned off and the program is ended.
130	Manual pulse generator	Normal	Not start	3	<ul style="list-style-type: none"> • 1 pulse input magnification of the 1 axis and 2 axes is set up. • 1 axis is controlled with P1, and set up to control 2 axes with P2, and Manual pulse generator enable flag of P1, P2 is turned on. • When except for PX2 : OFF, PX1 : ON (Manual pulse generator mode), Manual pulse generator enable flag of P1, P2 is turned off, and a program is ended.
140	Home position return	Normal	Not start	3	<ul style="list-style-type: none"> • "K140 : The home position return of 1 axis" is started when PX3 is on, "K141 : The home position return of 2 axes" is started when PX4 is on. • PX2 : ON, PX1 : The program is ended when they become to except for off (Home position return mode).
150	Programming operation	Normal	Not start	3	<ul style="list-style-type: none"> • When PX3 detects OFF to ON, after positioning of 1 axis, standing by for 1000[ms] and positioning of 2 axes is executed. • When PX4 turn on, after positioning of linear interpolation in-position check is executed, positioning of axis No. 1, 2 linear interpolation is executed at a double speed in the opposition direction, and it stand by until PX4 turned off. • PX2 : ON, PX1 : The program is fended when they become to except for ON (Programming operation mode).

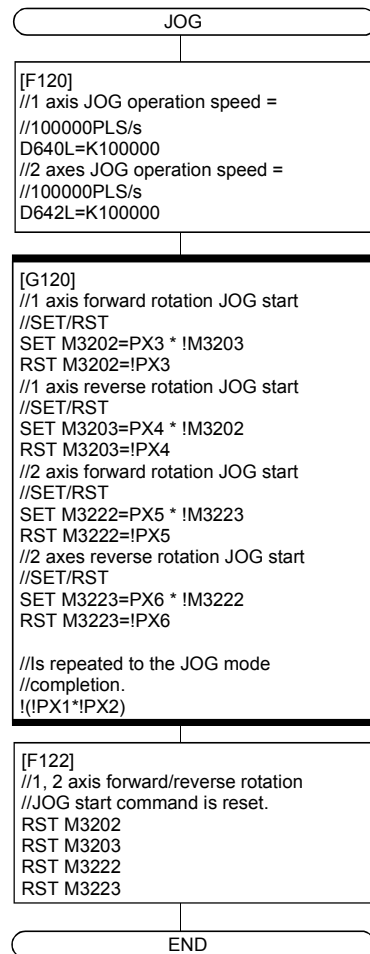
(a) No.20 : Main



(b) No.110 : Motion control



(c) No.120 : JOG



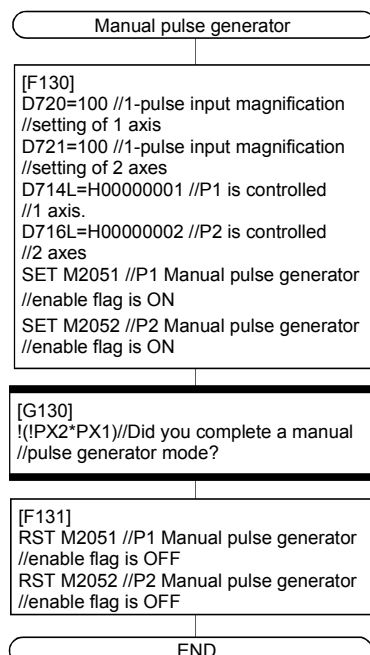
When each signal of PX3 to PX6 is turned on/off, which the correspondences JOG command device is SET/RST. It makes forward rotation JOG start of the same axis and a reverse rotation JOG start from making turned on at the same time.

Signal name	Correspond with JOG command device
PX3	M3202(1 axis forward rotation JOG)
PX4	M3203(1 axis reverse rotation JOG)
PX5	M3222(2 axis forward rotation JOG)
PX6	M3223(2 axis reverse rotation JOG)

(Note) : The ON/OFF distinction of each signal can be described with Y/N transition. But, processing time can be shortened more the number of steps when it was described as the following in the case of the processing which could be described only with SET=/RST= because it is made low.

Forward rotation/reverse rotation JOG status of 1, 2 axis is turned off at the time of the JOG mode completion not to continue a JOG movement after it moves to other mode of the safety.

(d) No.130 : Manual pulse generator

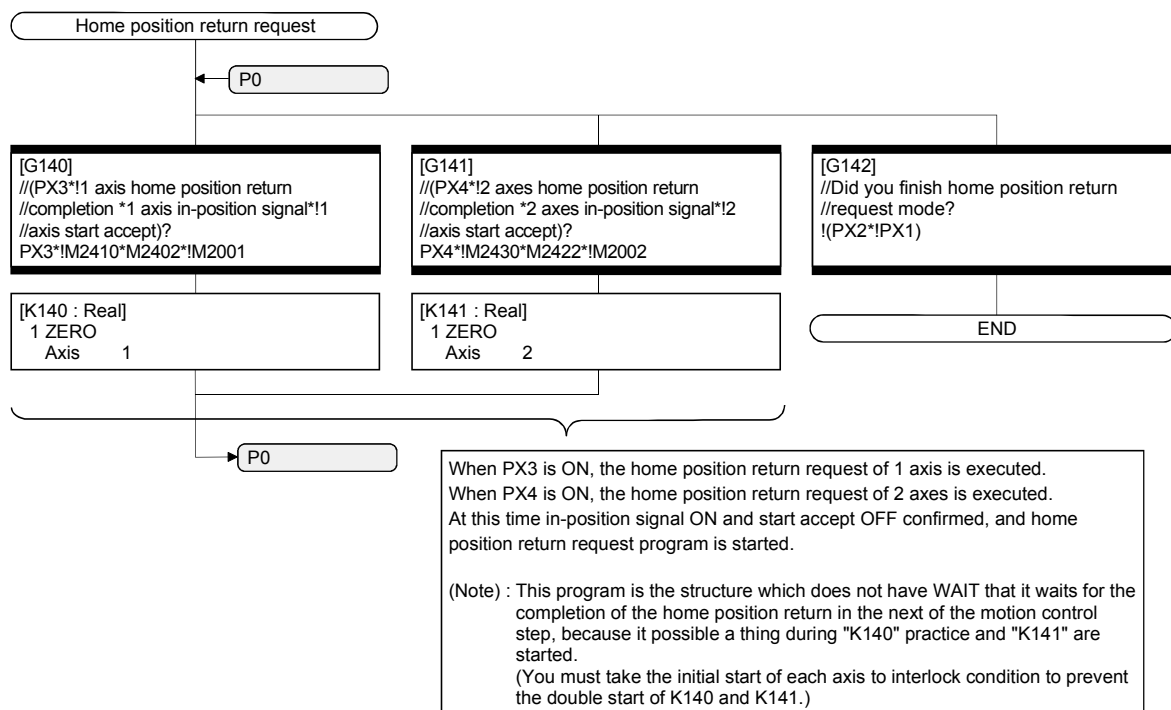


The setup of the following is executed to do manual pulse generator operation of P1 with 1 axis/P2 with 2 axis.

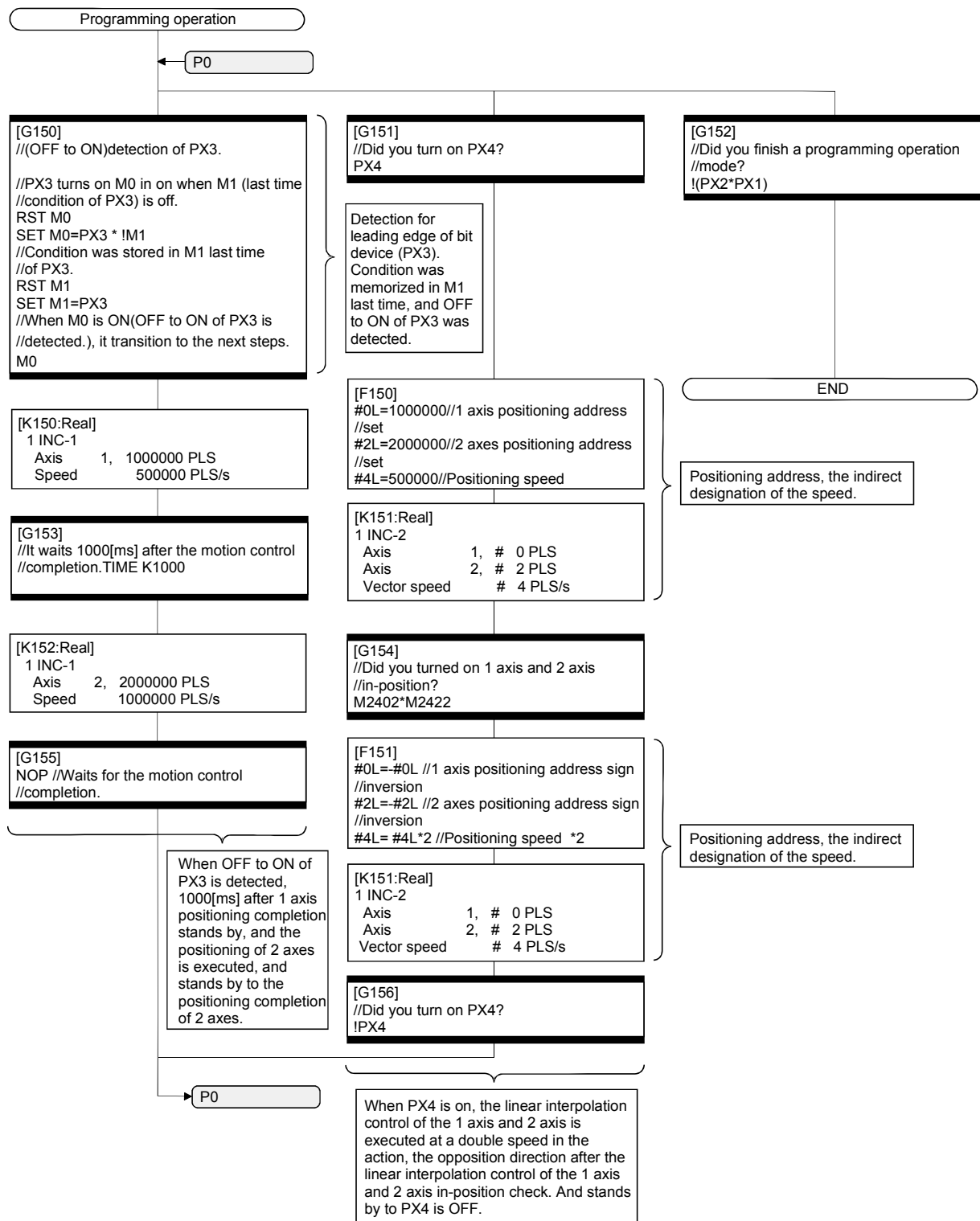
- Setting of 1-pulse input magnification of the 1 axis and 2 axis.
- Manual pulse generator axis No. setting register is setup to control of P1 with 1 axis/P2 with 2 axis.
- Manual pulse generator axis enable flag of P1, P2 is turned on.

1, 2 axis Manual pulse generator enable flag turned off at the time of the JOG mode completion not to continue a manual pulse generator operation after it moves to other mode of the safety.

(e) No.140 : Home position return

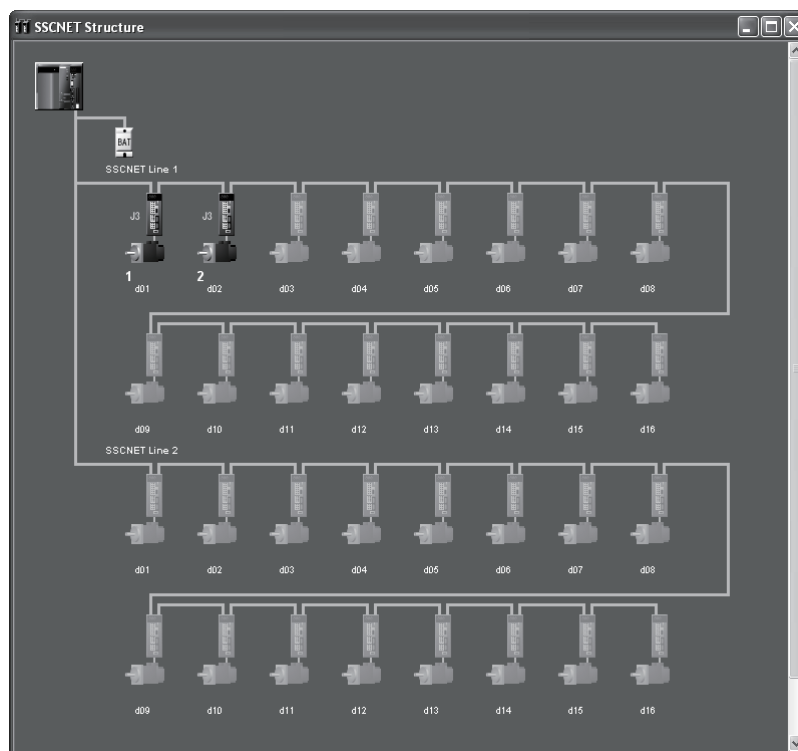
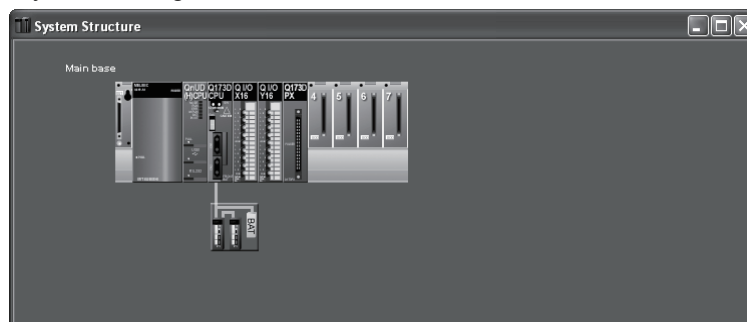


(f) No.150 : Programming operation



(3) System setting data of the Motion CPU

System setting is shown below.



(a) Module setting

1) Motion module setting

Manual pulse generator interface module (Q173DPX : Slot 3)

Axis No.	Description
P1	Manual pulse generator, Synchronous encoder (INC)
P2	Manual pulse generator, Synchronous encoder (INC)
I/O response time	0.4[ms]

2) PLC module setting

Module type	Points	Occupied I/O No.	Base	Slot No.	I/O response time
Input	16	000-00F	Main Base	1	10[ms]
Output	16	010-01F	Main Base	2	

(b) Basic setting

Basic Setting

Base Setting | **Multiple CPU Setting** | System Basic Setting

No. of CPU(*)
2 module(s)
Please set the number of Multiple CPU.

Operating Mode(*)
Error operation mode at the stop of CPU
☒ All station stop by stop error of CPU1
☒ All station stop by stop error of CPU2
☒ All station stop by stop error of CPU3
☒ All station stop by stop error of CPU4

Multiple CPU high speed transmission area setting

CPU	Points(k)	CPU specific send range(*)			Automatic refresh Points
		Points	Start	End	
No.1	7	7168	G10000	G17167	0
No.2	7	6468	G10000	G16467	700
No.3					
No.4					

Total 14k Points ☐ Advanced settings(*)
 The total number of points is up to 14k.
 Push the automatic refresh button to set the automatic refresh of the send range.
 Automatic refresh setting

Multiple CPU synchronous startup setting
 (*) Settings should be set as same when using multiple CPU.

OK Cancel

Basic Setting

Base Setting | Multiple CPU Setting | **System Basic Setting**

Operation Cycle
☐ 0.4ms
☐ 0.8ms
☐ 1.7ms
☐ 3.5ms
☐ 7.1ms
☐ 14.2ms
☒ Automatic Setting

Operation at STOP to RUN
☒ M2000 is turned on by switching from STOP to RUN.
☐ M2000 is turned on by switching from STOP to RUN and setting 1 in the set register.

Forced Stop
☒ Nothing ☐ X(IPX) ☐ M
 { 0 to 1FFF }

Latch Range

	Sym.	Device Range	Latch(1) Start	Latch(1) End	Latch(2) Start	Latch(2) End
Internal relay	M	0 to 8191				
Link relay	B	0 to 1FFF				
Annunciator	F	0 to 2047				
Data register	D	0 to 8191				
Link register	W	0 to 1FFF				

Latch(1): It is possible to clear using the remote operation (latch clear(1), latch clear(1)X[2]).
 Latch(2): It is possible to clear using the remote operation (latch clear(1)X[2]).

OK Cancel

1) Multiple CPU setting

Setting items	Description
No. of CPU	2 modules
Operating mode	All station stop by stop error of CPU 1/2
Multiple CPU synchronous startup setting	Set CPU No. 1/2 to synchronous startup

2) Multiple CPU high speed transmission area setting

CPU	Points (k)	CPU specific send range			
		User setting area			Automatic refresh Points
		Points	Start	End	
No.1	7	7168	G10000	G17167	0
No.2	7	6468	G10000	G16467	700
No.3					
No.4					

3) Automatic refresh setting

a) CPU No.1

Setting No.	Automatic refresh		
	Points	Start	End
1			
2			
3			

b) CPU No.2

Setting No.	Automatic refresh		
	Points	Start	End
1	20	M2000	M2319
2	40	M2400	M3039
3	640	D0	D639

4) System basic setting

Setting items	Description
Operation cycle	Automatic Setting
Operation at STOP to RUN	M2000 is turned on by switching from STOP to RUN
Forced stop	PX0

5) Latch range setting

Item	Symbol	Latch (1)		Latch (2)	
		Start	End	Start	End
Internal relay	M				
Link relay	B				
Annunciator	F				
Data register	D				
Link register	W				

Latch (1) : It is possible to clear using remote operation (latch clear (1), latch clear (1) (2)).

Latch (2) : It is possible to clear using remote operation (latch clear (1) (2)).

(4) Parameter setting of the QnUD(H)CPU No.1

Multiple CPU settings

No. of PLC (*)
No. of PLC: 2

Host CPU number
No specification

Operating mode (*)
Error operation mode at the stop of PLC
☒ All station stop by stop error of PLC1
☒ All station stop by stop error of PLC2
☒ All station stop by stop error of PLC3
☒ All station stop by stop error of PLC4

Multiple CPU synchronous startup setting (*)
Target PLC
☒ No.1
☒ No.2
☒ No.3
☒ No.4

(*)Settings should be set as same when using multiple CPU.

Online module change(*)
☐ Enable online module change with another PLC.
 When the online module change is enabled with another PLC, I/O status outside the group cannot be taken.

I/O sharing when using Multiple CPUs (*)
☐ All CPUs can read all inputs
☐ All CPUs can read all outputs

Multiple CPU high speed communication area setting | Communication area setting (refresh) [v]
☒ Use multiple CPU high speed communication

PLC	point(K)	I/O No.	User setting area			Auto refresh		
			point	Start	End	point	Start	End
No.1	7	U3E0	7168	G10000	G17167	0	---	---
No.2	7	U3E1	6468	G10000	G16467	700	G16468	G17167
No.3								
No.4								

Advanced settings(*)
 Total: 14K points

The total number of points is up to 14K.

PC parameter item		Description							
1	No. of PLC	2 modules							
2	Operating mode	All station stop by stop error of PLC1/PLC2							
3	Multiple CPU synchronous startup	Check the PLC No.1/PLC No.2							
4	I/O sharing when using Multiple CPUs	Check the all CPUs can read all inputs							
		Not check the all CPUs can read all outputs							
5	Multiple CPU high speed communication area setting	Use multiple CPU high speed communication							
		PLC	CPU specific send range						
			User setting area				Auto refresh		
			point (K)	I/O No.	point	Start	End	point	Start
		CPU No.1	7	U3E0	7168	G10000	G17167	0	—
CPU No.2	7	U3E1	6468	G10000	G16467	700	G16468	G17167	
6	Auto refresh setting	• PLC No1							
		No.	Auto refresh			CPU specific send range			
			point	Start	End	Start	End		
		1	—	—	—	—	—		
		2	—	—	—	—	—		
		3	—	—	—	—	—		
		• PLC No.2							
		No.	Auto refresh			CPU specific send range			
			point	Start	End	Start	End		
		1	20	M2000	M2319	G16468	G16487		
2	40	M2400	M3039	G16488	G16527				
3	640	D0	D639	G16528	G17167				

APPENDIX 2.2 Continuation execution example at the subroutine re-start by the Motion SFC program

(1) Explanation of the operation

This is the program example which execute continuously from the motion control step which stopped on the way when it re-started after stopping the subroutine program with the clear step during the motion control is running.

The servo is turned on by the forced stop release and the positioning control of the 2 axes liner interpolation is executed when PX4 is ON in this program. One cycle operation is completed after confirmation that PX4 became OFF. When the forced stop is executed during the positioning operating, the positioning operation is interrupted and the servomotor is stopped. It is resumed from the interrupted positioning operation when the forced stop was released next.

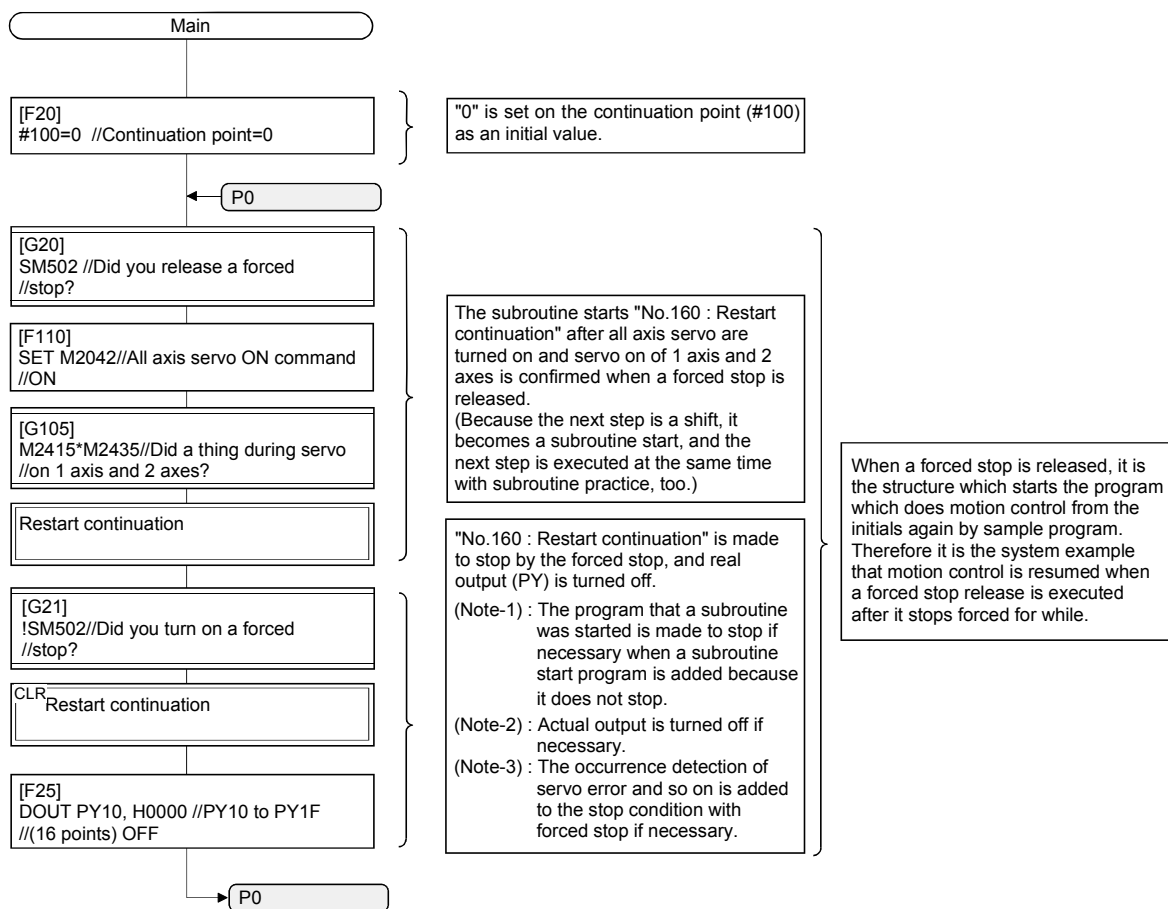
Continuation execution of the subroutine re-start is executed by this program example by the following processing.

- (a) While motion control with the subroutine is executed, it is memorized whether the positioning of which motion control step was completed in the user device.
- (b) The subroutine re-start is resumed from the motion control step of stopping the information memorized by the above (a).
- (c) A motion control step should locate absolute to cope with it when it is resumed after it stops on the way of the positioning.
- (d) A positioning complete signal (M2401+20n) is used for the decision, whether servomotor is stopped during the positioning.

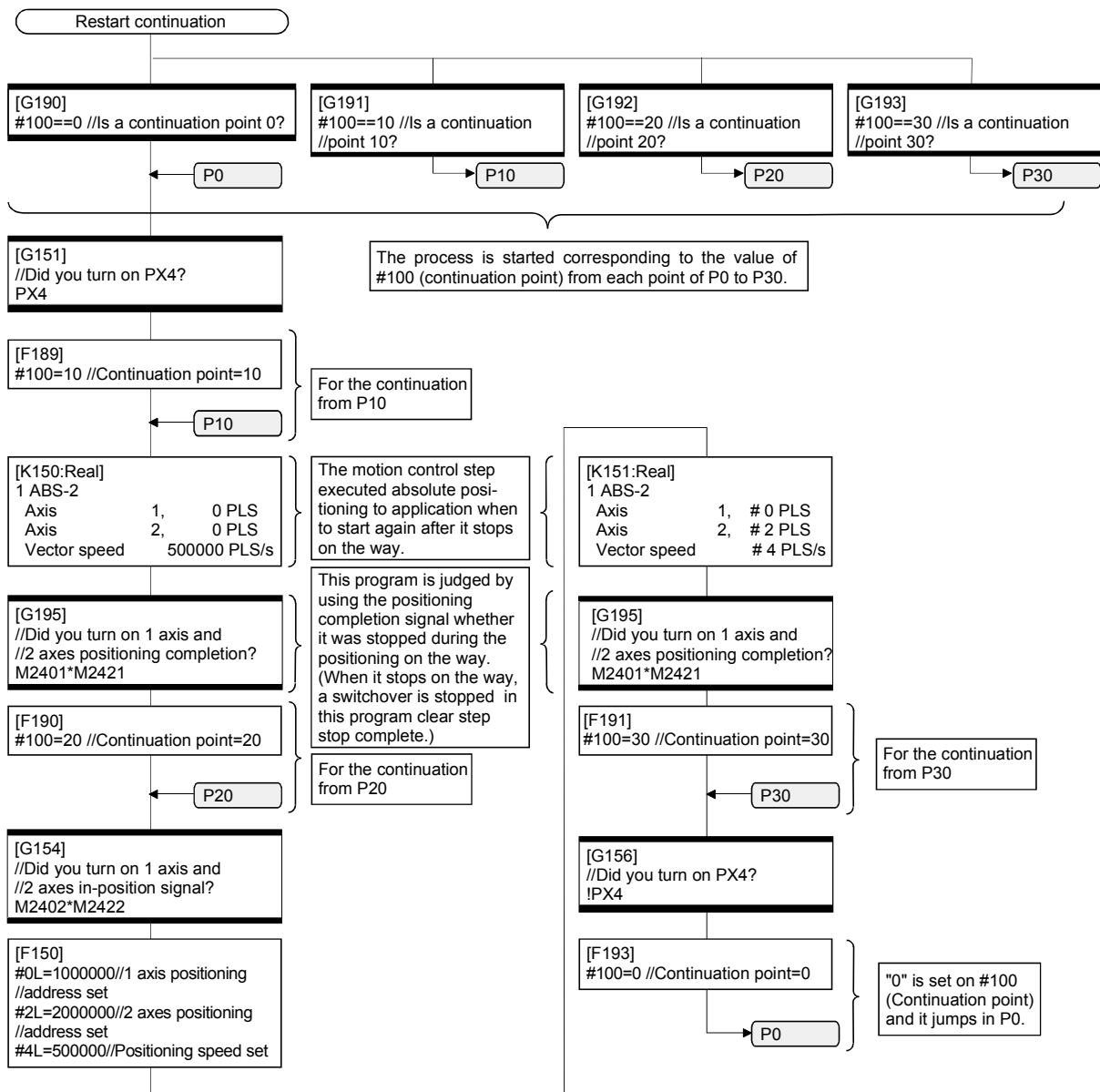
(2) Contents of processing the Motion SFC program
Motion SFC program list

No.	Program name	Task	Automatic operation	Number of connective transitions	Contents of processing										
20	Main	Normal	Start	3	<ul style="list-style-type: none">• This program starts automatically at the time of RUN of Q173DCPU, and it is always executed.• "0" is set on the continuation point (#100 : user device) as an initial value.• The subroutine starts a "No.160 : Re-start continuation" after all axes servo are turned on and servo on of 1 axis and 2 axes is confirmed when a forced stop is released.• "No.160 : Re-start continuation" is stopped at the time of the forced stop, and actual output (PY) is turned off.										
160	Restart continuation	Normal	Not start	3	<p>(1) This program jumps corresponding to the value of the continuation point (#100) of the following (2) 1) to 9).</p> <table><tr><th>#100</th><th>Jump destination</th></tr><tr><td>0</td><td>Following (2) 1)</td></tr><tr><td>10</td><td>Following (2) 3)</td></tr><tr><td>20</td><td>Following (2) 5)</td></tr><tr><td>30</td><td>Following (2) 8)</td></tr></table> <p>(2) The following motion control is executed.</p> <ol style="list-style-type: none">1) This program stands by until PX4 is turned on.2) "10" is set on continuation point (#100).3) 1 axis, 2 axes are located in (0,0) in the linear control (absolute 2 axes positioning).4) Positioning completion signal on of 1 axis, 2 axes is confirmed, and "20" is set on the continuation point (#100).5) In-position on of 1 axis and 2 axes is confirmed.6) 1 axis, 2 axes are located in (1000000, 2000000) in the linear control (absolute 2 axes positioning).7) Positioning completion signal on of 1 axis, 2 axes is confirmed, and "30" is set on the continuation point (#100).8) This program stands by until PX4 is turned off.9) "0" is set on continuation point (#100).	#100	Jump destination	0	Following (2) 1)	10	Following (2) 3)	20	Following (2) 5)	30	Following (2) 8)
#100	Jump destination														
0	Following (2) 1)														
10	Following (2) 3)														
20	Following (2) 5)														
30	Following (2) 8)														

(a) No.20 : Main



(b) No.160 : Restart continuation



APPENDIX 2.3 Continuation execution example after the stop by the Motion SFC program

(1) The explanation of the operation

The program example that the Motion SFC program is stopped by external input signal ON for the forced stop from the input module, and it is executed continuously by external signal OFF for the stop is shown below.

The servo is turned on by the forced stop release and the positioning control of the 2 axes linear interpolation is executed when PX4 is ON in this program. One cycle operation is completed after confirmation that PX4 became OFF. When PX5 turns ON during the positioning operating, the positioning operation is stopped by the stop instruction and it is resumed from the interrupted positioning operation at turning PX5 on. The transition to the next step is not executed during PX5 is ON in the WAIT transition.

When the forced stop is executed during the positioning operating, the positioning operation is interrupted and the servomotor is stopped. It is resumed from the interrupted positioning operation when the forced stop was released next.

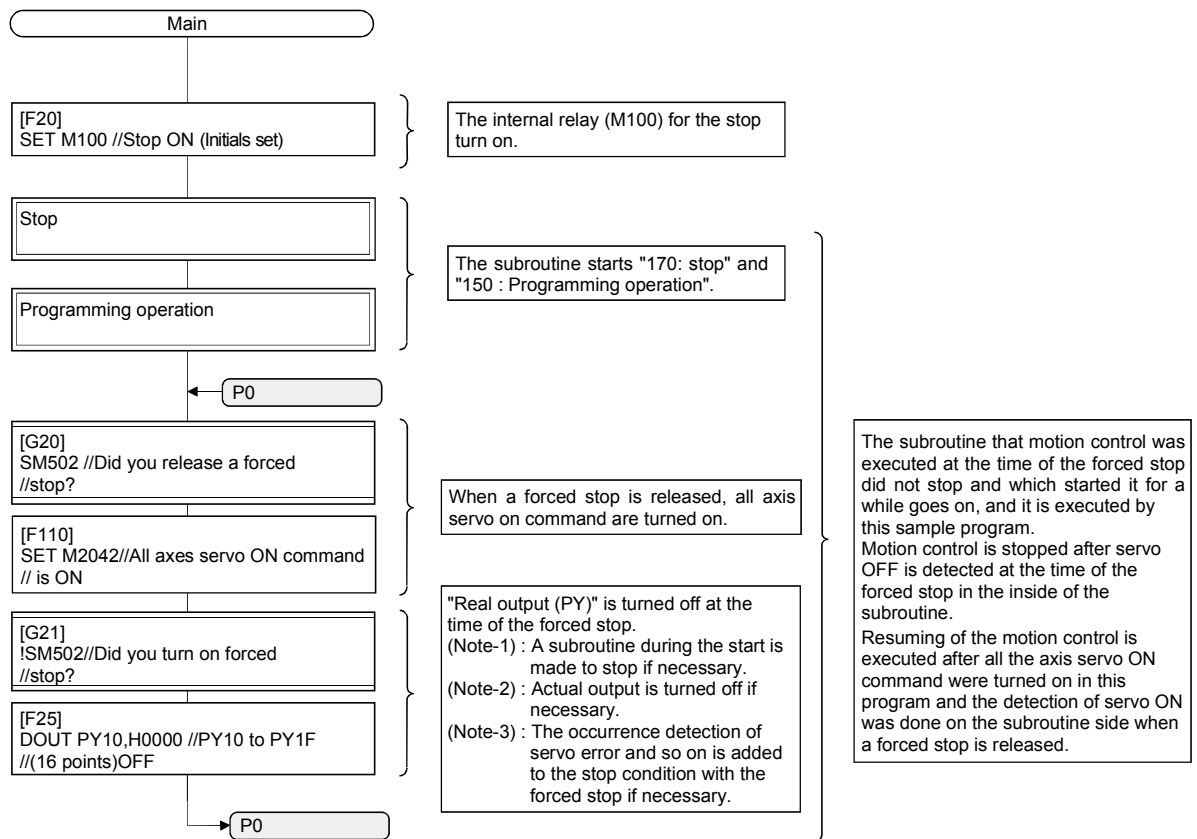
Continuation execution of the stop and stop after is executed by this program example by the following processing.

- (a) While PX5 turns it on, it is made to turn on a stop command (M3200+20n) and an internal relay (M100) for the stop.
- (b) While PX5 turns it off, it is made to turn off a stop command (M3200+20n) and an internal relay (M100) for the stop.
- (c) A motion control step does absolute position to cope with it when it is resumed after it stops on the way of the positioning.
- (d) A positioning completion signal (M2401+20n) is used for the decision whether it is stopped during the positioning on the way.
- (e) The motion control step is resumed after it waits to turn it off, when it was stepped during positioning.
- (f) "The internal relay (M100) for the stop turn off." is substituted for the WAIT transition condition that you must stop.

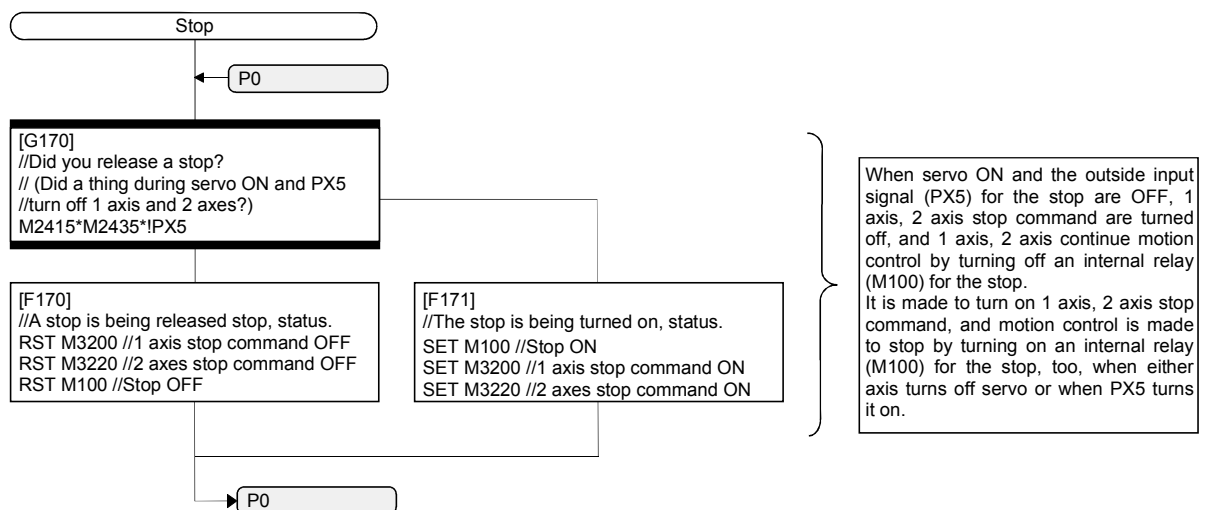
(2) Contents of processing SFC program
SFC program list

No.	Program name	Task	Automatic operation	Number of connective transitions	Contents of processing
20	Main	Normal	Start	3	<ul style="list-style-type: none"> • This program starts automatically at the time of RUN of Q173DCPU, and it is always executed. • The initials condition of the internal relay (M100) for the stop is turned on. • The subroutine starts "No.170 : Stop". • The subroutine starts "No.150 : Programming operation". • When an forced stop is released, all axes servo are turned on. • Turns off actual output (PY) at the time of the forced stop.
170	Stop	Normal	Not start	3	<p>(1) When a stop input signal (PX5) from the input unit is off, the treatment of the following (2) is executed, and 1 axis and 2 axes executed the following (3) during servo on in the case of the one except for it.</p> <p>(2) 1 axis and 2 axes stop command are turned off, and an internal relay (M100) for the stop is turned off.</p> <p>(3) 1 axis and 2 axes stop command are turned on, and an internal relay (M100) for the stop is turned on.</p>
150	Program operation	Normal	Not start	3	<p>(1) The following motion control is executed.</p> <ol style="list-style-type: none"> 1) This program stands by until PX4 is turned on. 2) 1 axis and 2 axes are located in (0,0) in the linear interpolation control (absolute 2 axes positioning). 3) Positioning completion signal on of 1 axis and 2 axes are confirmed. 4) In-position on of 1 axis and 2 axes are confirmed. 5) 1 axis and 2 axes are located in (1000000, 2000000) in the linear control (absolute 2 axes positioning). 6) Positioning completion signal on of 1 axis and 2 axes are confirmed. 7) This program stands by until PX4 is turned off. <p>(2) When a positioning completion signal of the above (1) 3) and 6) is off, it waits to turn off, and (When a positioning was suspended on the way.) execute the motion control step (1) 2) or 5) again.</p> <p>(3) Until an internal relay (M100) for the stop turns it on, it does not move to the next step of the above (1) 1) and 7).</p>

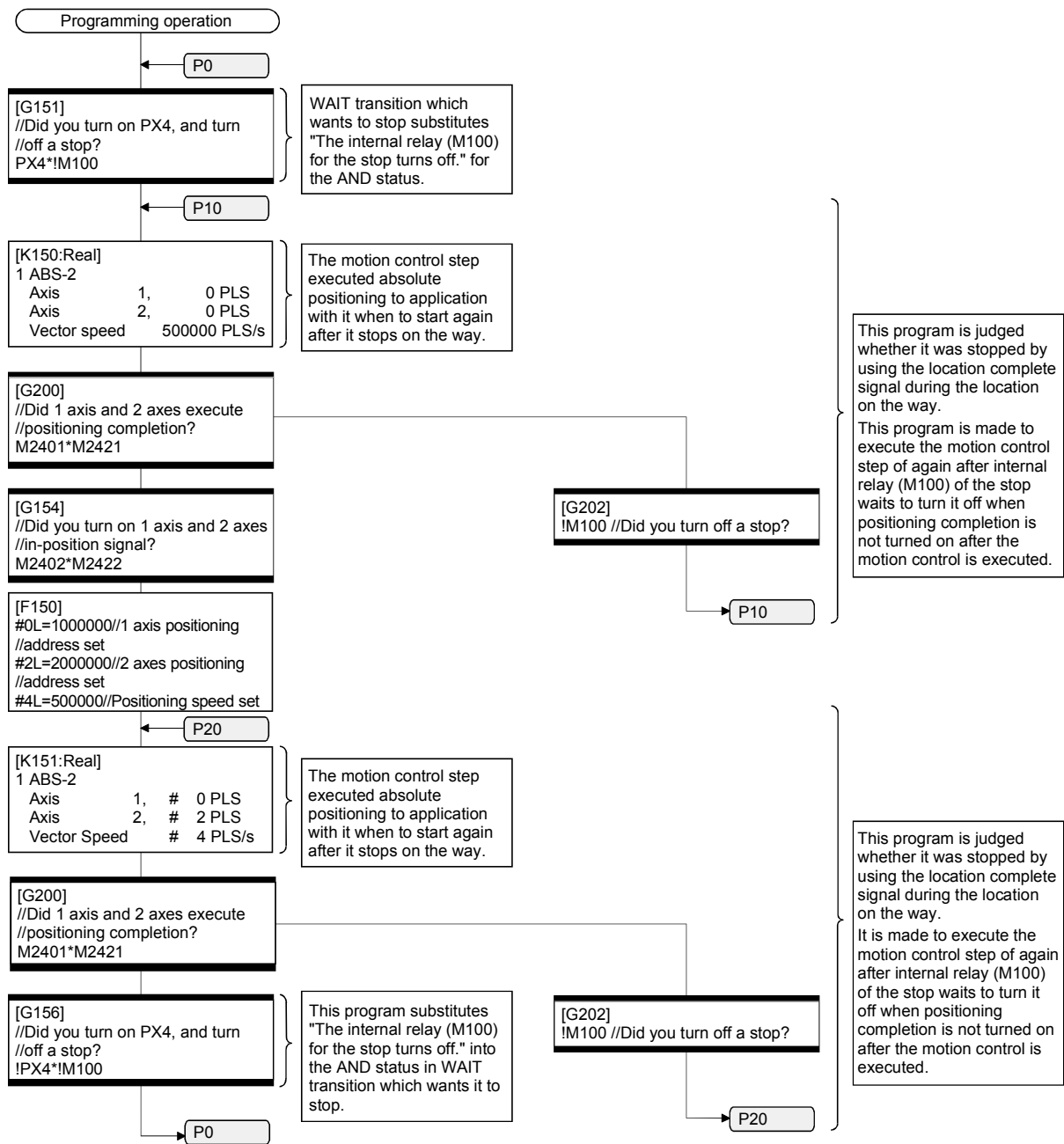
(a) No.20 : Main



(b) No.170 : Stop



(c) No.150 : Programming operation



MEMO

[illegible]

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

Note that an installation period of less than one year after installation in your company or your customer's premises or a period of less than 18 months (counted from the date of production) after shipment from our company, whichever is shorter, is selected.

[Gratis Warranty Range]

(1) Diagnosis of failure

As a general rule, diagnosis of failure is done on site by the customer.

However, Mitsubishi or Mitsubishi service network can perform this service for an agreed upon fee upon the customer's request.

There will be no charges if the cause of the breakdown is found to be the fault of Mitsubishi.

(2) Breakdown repairs

There will be a charge for breakdown repairs, exchange replacements and on site visits for the following four conditions, otherwise there will be a charge.

- 1) Breakdowns due to improper storage, handling, careless accident, software or hardware design by the customer
- 2) Breakdowns due to modifications of the product without the consent of the manufacturer
- 3) Breakdowns resulting from using the product outside the specified specifications of the product
- 4) Breakdowns that are outside the terms of warranty

Since the above services are limited to Japan, diagnosis of failures, etc. are not performed abroad.

If you desire the after service abroad, please register with Mitsubishi. For details, consult us in advance.

2. Exclusion of Loss in Opportunity and Secondary Loss from Warranty Liability

Mitsubishi will not be held liable for damage caused by factors found not to be the cause of Mitsubishi; opportunity loss or lost profits caused by faults in the Mitsubishi products; damage, secondary damage, accident compensation caused by special factors unpredictable by Mitsubishi; damages to products other than Mitsubishi products; and to other duties.

3. Onerous Repair Term after Discontinuation of Production

Mitsubishi shall accept onerous product repairs for seven years after production of the product is discontinued.

4. Delivery Term

In regard to the standard product, Mitsubishi shall deliver the standard product without application settings or adjustments to the customer and Mitsubishi is not liable for on site adjustment or test run of the product.

5. Precautions for Choosing the Products

- (1) These products have been manufactured as a general-purpose part for general industries, and have not been designed or manufactured to be incorporated in a device or system used in purposes related to human life.
- (2) Before using the products for special purposes such as nuclear power, electric power, aerospace, medicine, passenger movement vehicles or under water relays, contact Mitsubishi.
- (3) These products have been manufactured under strict quality control. However, when installing the product where major accidents or losses could occur if the product fails, install appropriate backup or failsafe functions in the system.
- (4) When exporting any of the products or related technologies described in this catalogue, you must obtain an export license if it is subject to Japanese Export Control Law.

MOTION CONTROLLER Qseries
SV13/SV22 Programming Manual(Motion SFC)
(Q173DCPU/Q172DCPU)



HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN

MODEL	Q173D-P-SV13/22-SFCE
MODEL CODE	1XB929
IB(NA)-0300135-A(0801)MEE	

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.